Department of Computer Science
Series of Publications A
Report A-2021-4

Computational Understanding, Generation and Evaluation of Creative Expressions

Khalid Alnajjar

Doctoral dissertation, to be presented for public discussion with the permission of the Faculty of Science of the University of Helsinki, in Auditorium B123, Exactum, Pietari Kalmin katu 5, on the 22nd of March, 2021 at 12:00 o'clock. The defence is also open for the audience through remote access.

University of Helsinki
Finland

Supervisor
Hannu Toivonen, University of Helsinki, Finland

Pre-examiners
Josep Blat, Universitat Pompeu Fabra, Spain
Tapio Salakoski, University of Turku, Finland

Opponent
Pablo Gervás, Universidad Complutense de Madrid, Spain

Custos
Hannu Toivonen, University of Helsinki, Finland

Faculty Representative
Teemu Roos, University of Helsinki, Finland

Contact information

Department of Computer Science
P.O. Box 68 (Pietari Kalmin katu 5)
FI-00014 University of Helsinki
Finland

Email address: info@cs.helsinki.fi
URL: https://cs.helsinki.fi/
Telephone: +358 2941 911

ISSN 1238-8645
ISBN 978-951-51-7145-0 (paperback)
ISBN 978-951-51-7146-7 (PDF)
Helsinki 2021
Unigrafia

Computational Understanding, Generation and Evaluation of Creative Expressions

Khalid Alnajjar

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
khalid.alnajjar@helsinki.fi
https://www.khalidalnajjar.com/
https://www.rootroo.com/

PhD Thesis, Series of Publications A, Report A-2021-4
Helsinki, March 2021, 56 + 91 pages
ISSN 1238-8645
ISBN 978-951-51-7145-0 (paperback)
ISBN 978-951-51-7146-7 (PDF)

Abstract

Computational creativity has received a good amount of research interest in generating creative artefacts programmatically. At the same time, research has been conducted in computational aesthetics, which essentially tries to analyse creativity exhibited in art. This thesis aims to unite these two distinct lines of research in the context of natural language generation by building, from models for interpretation and generation, a cohesive whole that can assess its own generations.

I present a novel method for interpreting one of the most difficult rhetoric devices in the figurative use of language: metaphors. The method does not rely on hand-annotated data and it is purely data-driven. It obtains the state of the art results and is comparable to the interpretations given by humans. We show how a metaphor interpretation model can be used in generating metaphors and metaphorical expressions.

Furthermore, as a creative natural language generation task, we demonstrate assigning creative names to colours using an algorithmic approach that leverages a knowledge base of stereotypical associations for colours. Colour names produced by the approach were favoured by human judges to names given by humans 70% of the time.

A genetic algorithm-based method is elaborated for slogan generation. The use of a genetic algorithm makes it possible to model the generation of text while optimising multiple fitness functions, as part of the evolutionary process, to assess the aesthetic quality of the output. Our evaluation indicates that having multiple balanced aesthetics outperforms a single maximised aesthetic.

From an interplay of neural networks and the traditional AI approach of genetic algorithms, we present a symbiotic framework. This is called the master-apprentice framework. This makes it possible for the system to produce more diverse output as the neural network can learn from both the genetic algorithm and real people.

The master-apprentice framework emphasises a strong theoretical foundation for the creative problem one seeks to solve. From this theoretical foundation, a reasoned evaluation method can be derived. This thesis presents two different evaluation practices based on two different theories on computational creativity. This research is conducted in two distinct practical tasks: pun generation in English and poetry generation in Finnish.

Computing Reviews (2012) Categories and Subject Descriptors:

Computing methodologies → Artificial intelligence → Natural language processing → Natural language generation
Computing methodologies → Machine learning → Machine learning approaches → Bio-inspired approaches → Genetic algorithms
General and reference → Cross-computing tools and techniques → Evaluation

General Terms:
Algorithms, Languages, Experimentation

Additional Key Words and Phrases:
Computational Linguistic Creativity, Natural Language Generation, Evaluation of Creative Systems, Artificial Neural Networks, Genetic Algorithms, Computational Methods

Acknowledgements

First and foremost, I would like to thank my supervisor Professor Hannu Toivonen for his support, insightful discussions and constructive feedback that bolstered my academic research and made this thesis possible.

I would like to give a special thanks to Dr Tony Veale for sparking my interest in the field of computational creativity, and for all the nice moments spent in Dublin and all the great opportunities I was given the chance to be part of. I would also like to thank everyone who has been involved in my PhD journey and had a positive impact on it, starting with all my co-authors (especially Ping Xiao; thanks for all the nice discussions and chats, including the ones outside the scope of work) and both previous and current members of the Discovery research group at the Department of Computer Science, all the way to researchers I have met during scientific conferences. Additionally, many thanks to the pre-examiners Josep Blat and Tapio Salakoski for reviewing the thesis and their helpful comments.

I truly appreciate the financial support received from the Academy of Finland's project Computational Linguistic Creativity (CLiC), Concept Creation Technology (ConCreTe) and Promoting the Scientific Exploration of Computational Creativity (PROSECCO) projects of EU FP7, Helsinki Institute for Information Technology (HIIT) and European Union's Horizon 2020 project Cross-Lingual Embeddings for Less-Represented Languages in European News Media (EMBEDDIA). With your support and financial aid, it was possible to focus on my research and participate in international high-quality academic events.

To my family, I will forever be grateful for having you as part of my life. My mother Zainab, father Ahmed Nasser, brother Mohammad and sister Najah, no words would be enough to express my gratitude for the emotional support, encouragement to pursue this path and all the things you have done that led me to this phase in life. I cannot thank you enough for being always by my side and for all the love you continuously provide. You are the best!

Mika Hämäläinen, you have been a great friend and it has always been

nice to brainstorm, discuss research topics and write scientific papers with you. So thank you! I would also like to thank Jack Rueter for great opportunities to work on low-resourced languages, I truly admire your devotion towards preserving endangered languages. I would like to thank Jörg Tiedemann, Jorma Laaksonen and Mikko Kurimo for the magnificent opportunity to continue my academic career as a post-doctoral researcher with the financial support from the Finnish Center for Artificial Intelligence (FCAI). Moreover, thanks to uncle Nabil for being supportive throughout my academic career, my friends Helmi Alhaddad, Soroush Atarod and the rest of my family members and friends who I have not mentioned by name.

Lastly, I would like to give my appreciation to the Palestinian Kunafeh, Qatayef and Ka'ak for always being there to cheer me up, especially when needed the most during stressful times.

Helsinki, March 2021
Khalid Alnajjar

Contents

Chapter 1

Introduction

Creativity is a trait that is fundamentally linked to being a human, and it has been approached by many philosophers over the course of time (cf. Gaut 2012). In the recent decade, a new paradigm for creativity has emerged taking the human into the realm of the computational. This new paradigm has then become known as computational creativity, which has been famously described as "the philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative" by Colton and Wiggins (2012).

Although computational creativity applications touch different fields of art ranging from music generation (Carnovalini and Rodà 2020) to paintings (Colton 2012), this thesis focuses on creativity in the context of natural language generation. Unlike a typical natural language generation task that focuses on producing linguistic form from some higher-level semantic representation communicating that particular representation (see Reiter 1994), creative natural language generation focuses on producing text in natural language that is aesthetically pleasing and has a degree of novelty to it.

Lately, the importance of the meaning conveyed by computationally creative text has been discussed in the theoretical work in the field (Hämäläinen and Honkela 2019). In addition to pleasing aesthetics, a computationally creative system is fundamentally communicating a meaning by its use of words. Regardless of how intentionally planned the meaning is by the computer, people will read meaning into anything expressed in a natural language.

In linguistics, meaning is usually divided into two categories: semantics and pragmatics. Semantics is understood as some sort of an absolute meaning of words, that is, what they mean without any further context. Whereas pragmatics takes the context more into account, and it requires

more interpretation as it enables indirect communication, where the meaning conveyed by a message is not the sum of the semantic meanings of the words it is expressed in.

Language can be used to communicate about a variety of things, such as facts, emotions, promises and orders (Searle 1969). In fact, language can be used to negotiate and shape the social reality around us through discourse (c.f. Foucault 1969). Thus languages are in constant dialogue with reality, describing and manipulating it.

Semantics have multiple times been solved to a great degree in the course of Natural Language Processing (NLP) research, from hand-authored WordNet (Miller 1998) and FrameNet (Baker, Fillmore, and Lowe 1998) to machine learned word embeddings such as word2vec (Mikolov et al. 2013) and BERT (Devlin et al. 2019). Pragmatics, on the other hand, have not been fully solved to a satisfying degree. One of the key focuses of my thesis is figurative language, which is a subset of pragmatics.

In contrast to literal language where the meaning is communicate directly, figurative language is a medium, like no other, for humans to communicate intricate messages that exceed the semantic meaning of the words alone. Figurative language is ubiquitous, we see it every day in our daily communications, news, advertisements, movies and any text that surrounds us. For example, Nokia's slogan "Connecting People" paints an image of mobile devices establishing intimate relations between people, in addition to providing a concrete means of communication between them.

With the power of figurative language, we can express complicated ideas, comparisons, experiences and emotions as presented in Nokia's slogan. Some example forms of figurative language include metaphors, similes, euphemisms and rhetorical questions. Successful decoding of a message expressed in figurative language in the mind of the recipient requires cultural understanding (c.f. Shao et al. 2019) or inter-subjectivity (Happé 1993). Hence, the interpretation of figurative language is an essential component of a system capable of computationally generating figurative language beyond the merely produced mundane surface form. A generative system that can interpret figurative language can also provide a reasoning for its output.

In this thesis we focus on computational creativity algorithms and models that deal with text in natural language from three core perspectives: 1) interpretation, 2) generation and 3) human evaluation. Paper I focuses solely on the interpretation of one type of figurative language, namely metaphor. Paper II presents an approach for natural language generation with a high level of perceived creativity, and Papers III-V present combined approaches where both interpretation of figurative language and generation

are simultaneously present in such a way that generation is informed by interpretation. Papers IV and V also have a strong theoretical grounding which makes the critical inspection of the extent of their creativity possible. When creativity is defined it becomes more approachable by scientific methods.

1.1 Creativity

Creativity is a momentous phenomenon in what it means to be a human. Unsurprisingly creativity has provoked philosophical interest for aeons starting from the ancient Greece (Asmis 1992). Pinpointing what creativity truly means is a difficult undertaking as it has meant different things in different times (see Gaut 2012). Furthermore, the essence of creativity differs from one culture and individual to another (Shao et al. 2019) due to its socially constructed nature (c.f. Moscovici 1961). Despite all the different theories, scholars seem to agree that creativity results in a product that is novel and has a value (Mumford 2003).

As Colton (2009) points out, computational creativity research can indeed benefit from understanding human creativity and the theories about it. However, Colton states that computational creativity research should not wait for research on understanding human creativity as we can approach the same phenomena from a different angle by building computational creativity systems. Therefore, these two paradigms can simultaneously work towards the same goal of understanding creativity.

In this section, we look at theories regarding human creativity and, following that, work conducted in the field of computational creativity. Theoretical understanding of creativity in modelling it computationally is crucial as argued in Papers IV and V.

1.1.1 Human Creativity

Despite the existence of philosophical takes on creativity starting from the Antiquities, which often explain creativity as something god-sent or divine. We dedicate this section to more modern theories. Of special interest are the theories that have been embraced in the computational creativity community.

An often cited theory in computational creativity is that of PPPP (4Ps) (Rhodes 1961). This theory explains creativity by four key components, 1) person, 2) process, 3) press and 4) product. The theory states that creativity should be studied from these four perspectives. Person refers to the individual that performs the creative act and it is to be understood

through the psychological traits, general intellect, habits etc of the individual. This raises the question of whether every individual can even be creative and to what degree they can be creative. Creativity does not exist in a vacuum as it is inherently social; therefore, the theory uses the notion of press to describe the relationship of the person with their environment.

In the 4Ps theory, process is the the path that one goes through to reach the final creative outcome. This is not limited to the skill set that makes a painter paint a painting but rather also covers aspects such as the motivations, internal thinking process, experiences and active engagement in solving the creative problem. The creative outcome is known as the product in the theory. The theory states that it is an expression of an idea and a particular idea can be incarnated into several different products.

Csikszentmihalyi (1997) has a similar view to that of Rhodes (1961), he states that creativity consists of three components, 1) culture, 2) person and 3) experts in the field. The theory highlights the interaction between the three components. Culture is considered as a set of symbolic rules and creativity occurs when a person introduces a novel concept to culture. Experts in the field are the judges that judge the quality and innovation of the concept introduced.

Moreover, Csikszentmihalyi distinguishes between two levels of creativity, which have later come to be known as little-c and big-c creativity, although he did not establish these terms himself (Merrotsy 2013). Little-c refers to creativity that consists of innovations that gain importance only on a personal level as they are not supported by the culture or experts in the field. Big-c creativity can only occur if the personal creativity is recognised by the two other components of creativity. This thinking is close to that of Boden (2004), but we will discuss her ideas in the following section that focuses on computational creativity.

Wallas (1926) has proposed a four-phased creativity model which aims to describe the process for reaching a creative outcome. These phases are: 1) preparation, 2) incubation, 3) illumination and 4) verification. Preparation is the phase where the problem is identified, and knowledge and information that are related to it are acquired and collected. During the incubation phase, one does not actively seek answers to the problem, but rather observes it from a distance to contemplate the problem. The third phase, i.e. illumination, marks the discovery of a solution, like an aha-moment that emerges suddenly. Finally, the viability of the solution is verified in the last stage.

Creativity can be assimilated with the idea of exploratory thought as expressed by Lerner and Tetlock (2003). When engaging in exploratory

thought, one tries to approach the problem from different angles and perspectives, and then pick the solution that suits the problem best. In contrast, confirmatory thought focuses on approaching the problem from a single point of view to test (confirm) whether the solution is applicable.

While there is no clear measurement for assessing creativity, one noteworthy take was conducted by Torrance (1962). Torrance proposed a test for assessing one's creativity based on a set of problem-solving and divergent thinking tests, where the performance on each test would be measured by four scales: 1) fluency; the number of apt ideas created, 2) flexibility; the number of unique categories these ideas could fall under, 3) originality; the novelty of the idea in comparison to other known answers, and 4) elaboration; the extent of details these ideas had. Usually, children in elementary schools were the focus of his tests, and tests could be verbal or figural.

1.1.2 Computational Creativity

In this section, I focus on theories on creativity from the computational point of view. The field of computational creativity has created a myriad of theoretical definitions for creativity of its own, which better take into account having something non-human, such as a computer, as a creative agent than the theories explaining human creativity.

One of the most foundational theories in the field of computational creativity was formulated by Boden (2004) and later revisited by the same author in Boden (2007). The theory introduces a simple dichotomy for the level of creativity; H- and P- creativity. H-creativity refers to historical creativity meaning that anything that is novel enough to make a big impact and is unlikely to have emerged easily elsewhere is considered historically creative. For instance, the invention of electricity is considered H-creative. P-creativity, on the other hand, is psychological creativity, which means that the invention is novel to the inventor himself, but is probably something a great many people have already thought of before. Solving a difficult puzzle or a riddle can be a P-creative act.

Another important aspect of Boden's work is that she identified three different kinds of creativity: exploratory, transformational and combinational. In exploratory creativity, the computational system explores a conceptual space finding new creative artefacts within that space. If the space gets changed into something different by the system, we are dealing with transformational creativity. Such a transition allows the system to explore completely new artefacts. Combinational creativity means producing new concepts by combining existing ones

Wiggins (2006) has proposed a framework for analysing the creativity

of computational systems by viewing creativity as a search problem and the level of a computer's creativity would be determined based on how it explores the search space to find good solutions in it. Wiggins defines the notion of a universe, which includes all artefacts; however, a system explores only a subset of it. In such a framework, a system could exhibit any of the three types of creativity mentioned by Boden (2004). If, for instance, a computer used to combine answers to reach new ones in the search space, combinational creativity would be exhibited. In case the machine explored the space by using some heuristics that comply with predefined rules, it would be considered to have exploratory creativity. The highest level of creativity, i.e. transformational, is attributed to a system if it is capable of altering its search space, its own rules for exploring the search space and/or criteria for assessing answers to find surprising, new and useful solutions.

Ritchie (2007) approaches creativity by listing a non-exhaustive set of criteria that take into account solely the artefact produced by computational means. The criteria are built around *novelty*, *quality* and *typicality* and the degree to which they are exhibited by the output produced by the system. This take does not set too strict a requirement for the creative process in place in the system, but rather focuses on the output and its relation to the *inspiring set*. The inspiring set can be compared to training data in machine learning, but it should be understood in a very broad sense as it also covers any artefacts that might inspire the results such as those the programmer is aware of, data recorded in knowledge bases and so on.

Colton (2008) proposes a so-called creative tripod framework to describe computational creativity. This theory is followed closely in Paper VI. According to this theory, creativity consists of three components: skill, imagination and appreciation. And all of these can be present in the three different parties engaging in the creative act: programmer, program and perceiver, each of which can contribute to the overall creativity of the system.

Skill refers to the capacity of the system to create a desirable artefact. A painter can paint a painting only if he knows how to move the brush along a canvas to form an illustration. Such a painter would be exhibiting skill, but if the paintings are unoriginal, repetitive and unimaginative, he lacks imagination. A creative system should thus be able to produce a lot of variety in its output. The last component, appreciation, means that the system should be able to assess its own creations in a meaningful way. A painter who lacks appreciation, cannot tell whether his paintings are good or bad, let alone why they are good or bad.

The theoretical background for Paper V is the one suggested by Colton,

Charnley, and Pease (2011). The theory describes creativity through the FACE model, consisting of framing, aesthetics, concepts and expressions. The theory states that each part of FACE can be divided into a ground-level creative act and how that act came to be. Expressions are the creative output produced by the system and concept refers to the creative program producing such expressions. Aesthetics are much like appreciation in the creative tripod, it is the capacity of the system in assessing its own creations on different parameters. Framing is probably the most peculiar part of this theory. It can be about providing a wider socio-cultural context to the expressions or an explanation for the creative decisions taken by the system etc.

Hämäläinen and Honkela (2019) identify that, in computational linguistic creativity in particular, the existing theoretical work tends to ignore the communicative use of natural language. People speak and write first and foremost to communicate meanings rather than solely to produce aesthetically pleasing utterances. In their theory, they assume a clear message to be conveyed by a goal-oriented dialog generation system, and creativity can take place on three different levels: in the message, in the context or in the communicative act. A system that is too creative can hardly communicate the desired meaning effectively, thus they identify a communicative-creative trade-off.

1.2 Contribution of the Thesis

This thesis consists of five publications describing novel computational methods for interpreting, generating and evaluating creative language. In this thesis, we also showcase means to assess the creativity of NLG systems for generating creative language by applying existing computational creativity evaluation theories.

In Paper I, an unsupervised method for interpreting metaphors based on word associations is presented. Papers II, III and IV present various natural language generation methods for producing expressions exhibiting linguistic creativity ranging from straightforward algorithms and traditional machine learning methods such, as genetic algorithms, to more modern approaches, namely neural networks. Combining these two machine learning paradigms into a dual-agent master-apprentice framework makes it possible to inspect the emergence of computational creativity between two systems.

Towards the last papers, Papers IV and V, the role of theoretical grounding, where creativity should be first defined on a theoretical level so that evaluation can be derived from that definition (Jordanous 2012), becomes

more important. Therefore, these two papers follow a theoretical foundation that is explicitly modelled by the computational approach and evaluated.

1.2.1 Original Publications

The publications included in this thesis are the following:

PAPER I - Meta4meaning: Automatic Metaphor Interpretation Using Corpus-Derived Word Associations. Xiao, P., Alnajjar, K., Granroth-Wilding, M., Agres, K., & Toivonen, H. (2016). In *Proceedings of The Seventh International Conference on Computational Creativity* (pp. 230–237). Paris, France: Sony CSL Paris.

The paper represents a novel unsupervised method for interpreting the meaning of nominal metaphors (i.e. metaphors in the format "NOUN is [a/n] NOUN") automatically by obtaining word associations and performing vector operations on both nouns. The notions developed in this paper were later used for metaphor generation in Papers III and V.

In this work, I have contributed to processing the corpus and building several different association matrices out of it. Furthermore, I have contributed to implementing the metaphor interpretation method and evaluating final results.

PAPER II - Grounded for life: creative symbol-grounding for lexical invention. Veale, T., & Alnajjar, K. (2016). Connection Science, 28(2), 139–154.

This paper presents a novel technical approach to naming colours automatically by mining stereotypical colour associations and leveraging them (along with their RGB colour mappings) to construct descriptive colour names.

The algorithmic approach proposed in this paper for naming colours was built by me. Additionally, I was in charge of conducting the human evaluation on a crowdsourcing platform. I also contributed in writing the paper.

PAPER III - Computational Generation of Slogans. Alnajjar, K. & Toivonen, H. (2020). Natural Language Engineering. *Natural Language Engineering*, 1-33.

This paper extends my MSc thesis work (Alnajjar 2019) in constructing metaphors based on their interpretations and generating advertising slogans automatically using genetic algorithms and multi-objective optimisations. This paper includes a more thorough evaluation and analysis than what was described in the MSc thesis.

The work described in this paper has been done by me with helpful discussions with my supervisor, the second author. In terms of implementation and evaluation, this paper entirely represents my own work. The paper has been written by me to a great extent, although my supervisor has contributed to the writing as well.

PAPER IV - A Master-Apprentice Approach to Automatic Creation of Culturally Satirical Movie Titles Alnajjar, K., & Hämäläinen, M. (2018). In *Proceedings of the 11th International Conference on Natural Language Generation* (pp. 274–283). Stroudsburg, PA: The Association for Computational Linguistics.

This paper establishes the so-called master-apprentice approach which consists of a computationally creative genetic algorithm and a sequence-to-sequence recurrent neural network-based model. In this approach, the master (genetic algorithms) teaches the apprentice (sequence-to-sequence neural model) to generate satirical movie title puns. The apprentice is also exposed to human annotated data. This framework has then been further expanded and studied in Hämäläinen and Alnajjar (2019c) and Paper V. One of the key notions is the strong theoretical grounding and emphasis on a reasoned evaluation.

We contributed equally to the implementation of the system, my co-author and I. We were both involved in processing the corpus and conducting the evaluation. Writing the paper was also a joint effort by both of the authors.

PAPER V - Let's FACE it. Finnish Poetry Generation with Aesthetics and Framing Hämäläinen, M., & Alnajjar, K. (2019). In *Proceedings of the 12th International Conference on Natural Language Generation* (pp. 290–300). Stroudsburg, PA: The Association for Computational Linguistics.

This paper takes the preliminary poem generation approach presented in Hämäläinen and Alnajjar (2019b) and incorporates it into a more theoretically motivated master-apprentice framework. The approach taken in this paper is motivated by existing theoretical work on poetry and the evaluation is conducted by using evaluation questions as concrete as possible.

The theoretical foundation and the evaluation setup were discussed with my co-author. We contributed equally to the implementation of the system and conducting the evaluation. Writing the paper was also a joint effort by both of the authors.

Chapter 2

Understanding Figurative Expressions

Figurative language is a form of indirect communication, where the meaning of a sentence cannot be derived by the individual semantic meanings of the words. Figurative language, and especially metaphors, have been widely studied in the field of natural language processing (NLP) from the point of view of detection, interpretation and generation (Rai and Chakraverty 2020; Veale and Li 2013; 2012; Galván et al. 2016; Alnajjar et al. 2017).

The goal of metaphor detection is to identify and recognise metaphorical expressions from literal ones whereas metaphor interpretation attempts to find the intended meaning behind the non-literal expression. The underlying idea of metaphor generation systems is to utilise the different theories on metaphor processing along with knowledge bases (hand written or automatically mined) to produce novel and apt metaphors.

This chapter is dedicated to the topic of metaphor interpretation. I begin by briefly describing the the related theories on processing metaphors. Thereafter, I describe existing work on computational methods for metaphor interpretation and, then, present our take on metaphor interpretation. In the last section of this chapter, I show how figurative language interpretation can be used to produce metaphors and utilised as part of an NLG algorithm to generate creative language.

2.1 Metaphor Processing

Two concepts are fundamental for a metaphor, which are a tenor and a vehicle (Richards 1936). Let's take a look at the common metaphor "time is money". In this metaphor, *time* is the tenor and *money* is the vehicle.

As a result of comprehending this expression figuratively, *valuable* (a well-known property of *money*) gets highlighted and attributed to *time* without indicating it plainly. In a metaphor, some properties of the vehicle get implicitly highlighted or attributed to tenor. Metaphors often have more than one interpretation and depending on the context and the two concepts, certain interpretations get highlighted stronger than others. For instance, *desirable* is a possible interpretation of the previous metaphor but *valuable* is a typical interpretation for it.

Multiple theories exist in the literature about metaphors, from how we comprehend them to what characteristics are exhibited by them and what makes a metaphor apt. A common view on metaphors states that deciphering the meaning of a metaphor is a comparison process in which the domains and properties of the vehicle are compared to those of the tenor to find out which of them make sense in the context of the tenor (Gentner 1983; Kirby 1997). The *salience imbalance theory* (Ortony 1993; Ortony et al. 1985) extends this view and states that metaphoricity arise due to the salience imbalance between the shared properties of the vehicle (highly salient) and the tenor (slightly salient), which causes them to be highlighted.

On the contrary, Wilks (1978) suggests that dissimilarities between the domains are what provoke conflicts in comprehending the metaphorical utterance, which is followed by a seek to a sensible interpretation. Another theory views metaphors as class-inclusion assertions that prompt seeing the tenor in the perspective of the vehicle (Davidson 1978; Glucksberg, McGlone, and Manfredi 1997) From a cognitive perspective, Lakoff and Johnson (2008) view metaphors as cognitive conceptual mapping from an abstract concept (i.e. tenor) to a well-known concrete concept (i.e. vehicle),and Black (1962) argues that metaphors are understood by virtue of the interaction between the metaphorical focus of the expression and the literal context it is conveyed in.

In terms of properties of metaphors and what makes them apt, *similarities* within and between the domain of the vehicle and that of the tenor are aspects humans consider when comprehending metaphors (Tourangeau and Sternberg 1981). They also indicate that the relation between the tenor and vehicle is *asymmetrical*; hence, the metaphor "A is B" highlights different properties than "B is A". Katz (1989) points out that *concrete* vehicles that are *semantically moderately distant* from the tenor result in apt metaphors.

Metaphors can be expressed linguistically in various forms. A nominal metaphor is a metaphor where the tenor is equated to the vehicle using the copula, e.g. "time is money". Adjectives (e.g. *creative* energy), verbs (e.g. *speaks* for itself) and adverbs (e.g. travelled *fluidly*) of the vehicle can be

used as well in constructing metaphorical expressions, and in some cases even prepositions could be used metaphorically (e.g. level up *in* life). In such cases, the vehicle may remain implicit.

Some expressions might convey a metaphor in a complex form such as multi-word expressions, compound word and throughout a long discourse. For instance, the slogan of *Diners Club*[1] "The international symbol for YES" where the company (tenor) is asserted to a checkmark or OK gesture (vehicle) to convey that their card is internationally accepted.

2.2 Computational Interpretation of Metaphors

The task of interpreting metaphors automatically has been studied largely (c.f. Rai and Chakraverty (2020) for a survey). Nonetheless, like any task related to understanding art and creativity, metaphor interpretation is an AI-complete problem and cannot be completely solved. Majority of metaphor interpretation approaches focus on interpreting nominal metaphors, including the work presented in Paper I. Nonetheless, researchers have tackled other forms (Rosen 2018; Shutova 2010; Shutova, Van de Cruys, and Korhonen 2012; Mohler, Tomlinson, and Bracewell 2013). In this section, I will describe related work on computational methods for interpreting nominal metaphors and explain the proposed method in this thesis.

Kintsch (2000; 2008) have proposed a method that employs Latent Semantic Analysis (LSA) to acquire vector representation of terms in a semantic space, where semantic similarities between two terms is measured as the cosine similarity of their vectors. The author have used such knowledge to approximate the Construction-Integration (CI) model, a psychological model of text comprehension (Kintsch 1988), to predict the metaphorical meaning in nominal metaphors. To obtain the meaning of a metaphor, a centroid of the tenor, vehicle and words most related to them in the semantic model is composed and compared to a set of terms (*landmarks* that have been defined by hand) using the cosine similarity to highlight the meaning of the metaphor. It is worth noting that this method does not produce metaphor interpretations directly but rather infers them from the term landmarks.

Terai and Nakagawa (2008; 2012) approach consists of two processes: 1) a categorization process and 2) a dynamic-interaction process. Their approach makes use of a probabilistic latent semantic indexing (PLSI) model based on the dependencies (and their frequencies) between nouns and adjectives and between nouns and verbs, where the adjectives and verbs are

[1] A payment card company.

considered as the attributes of the nouns. As a result, connections between nouns and attributes are encoded (called *latent classes*) which are then used as vector dimensions. The first step of the metaphor interpretation process (i.e. the categorization process) follows the method of Kintsch (2000) to produce a centroid vector of the metaphor based on the tenor, noun and terms most related to both concepts. This vector is then used to estimate the salience score of attributes in the latent classes for a given metaphor. In the next process, the dynamic-interaction process, attributes having a salience score above a certain threshold are selected and used as nodes (along with their salience scores) in a recurrent neural network. Once the network has been trained, attributes with the highest activation are considered as the meaning of the metaphor.

Veale and Li (2012) employed linguistic patterns (such as "NOUN_1 is [a/n] NOUN_2", "VERB+*ing* lika a NOUN" and "[a/n] ADJ NOUN") to harvest stereotypical associations for nouns and connections between these stereotypical associations (using "as ADJ_1 and ADJ_2 as") from Google n-grams and the web via Google Search API. To ensure building a high quality knowledge base, the authors have manually checked the mined associations. The authors provide a service called *Metaphor Magnet* that uses the retrieved knowledge to interpret and generate metaphors. The interpretation process looks at all the terms associated with the tenor and vehicle, and examines each as an interpretation by observing the overlapping properties between the three concepts. Terms that have a high overlapping ratio with the tenor and vehicle get highlighted and considered as interpretations of the metaphor.

Su, Tian, and Chen (2016) suggested a method for interpretation of metaphors based on the semantic properties of the tenor's and vehicle's domains. They have utilised a pre-trained word2vec model to interpret metaphors as follows. Given a nominal metaphor, the method retrieves properties salient in the domains of the tenor and vehicle from knowledge bases such as *Sardonicus* (Veale and Hao 2007), and only shared properties between the two domains are considered. Synonyms of each property are acquired from sources like *WordNet* (Miller 1995), and the interpretation score for a given property is calculated as the average semantic similarity of the property's synonyms to the tenor's domain in the pre-trained vector space. Properties with the highest interpretation score are considered as the meaning of the metaphor.

Similarly to the approach by Su, Tian, and Chen (2016), Rai et al. (2019) describe an approach for interpreting metaphors using a pre-trained word2vec model but from a different perspective, highlighting interpreta-

tions based on emotions they provoke. Properties for both tenor and vehicle are obtained from the semantic model (i.e. word2vec), in case they have a semantic similarity score within defined thresholds. Subsequently, six emotions are considered (namely *anger, fear, happiness, disgust, sadness* and *surprise*) (Cipresso, Serino, and Villani 2019). For each emotion, the property with the maximum semantic similarity to it and to the tenor is considered to stimulate the emotion and is picked as an interpretation.

In Paper I of this thesis, we tackle the same task of interpreting nominal metaphors. We propose a data-driven approach (called *Meta4meaning*) based on word associations collected from a web corpus. These associations, after applying lematization and omitting punctuation, are converted into semantic relatedness scores using the simple log-likelihood measure and normalising scores for each word using the L1-norm. When interpreting a metaphor, the approach takes into consideration shared adjectives and abstract nouns and verbs of both tenor and vehicle which are then ranked based on their relatedness scores. Multiple ranking measures were compared in the task of approximating existing metaphor comprehension theories (e.g. salience imbalance theory (Ortony 1993)). Based on the experiments, the best measure was a combination of the magnitude (product) of relatedness values to the tenor and vehicles, and the difference between the relatedness to the vehicle and the tenor (i.e. salience imbalance).

Meta4meaning is capable of interpreting metaphors if the tenor and vehicle occur in the corpus with other words adequately. To overcome this obstacle, an extension to *Meta4meaning* has been suggested by Alnajjar et al. (2017) where (a few) adjectival properties of rare concepts such as famous proper nouns are automatically expanded and weighted using automatically mined links between adjectives, allowing the method to produce interpretations to metaphors like "*Hillary Clinton* is a *cat*". Moreover, Bar, Dershowitz, and Dankin (2018) extended *Meta4meaning* by testing out different modifications such as using syntactic dependencies to obtain collocations and applying clustering. Their results show that applying semantic clustering to remove semantically duplicate features improves the interpretations. Nonetheless, *Meta4meaning* appears to surpass the extended method and achieve the state-of-the-art results.

Our approach, *Meta4meaning*, obtains the knowledge about related properties and terms of tenor and vehicle by an unsupervised corpus-driven method and, with this knowledge, a relatedness model is constructed that is used to interpret metaphors. These points, along with the novel way of ranking interpretations, distinguish our approach from the rest. Our evaluation conducted in Paper I shows that *Meta4meaning* suggests interpreta-

tions closer to gold-standard human interpretations than the ones produced by *Metaphor Magnet* (Veale and Li 2012) and that of Terai and Nakagawa (2008). Furthermore, some differences regarding the theories inspiring our and current work exist (c.f. Rai and Chakraverty (2020) for more details).

A core difference between our approach and the rest is the type of semantic relations that we focus on. Different types of relations can be extracted from a corpus, e.g. syntagmatic relations and paradigmatic relations (Rapp 2002). Syntagmatic relations capture words that co-occur together within a given boundary (e.g. the *dog* has *fur* or the *dog* is my *pet*) while paradigmatic relations capture words that share similar contexts and can substitute each other (e.g. *dog* and *cat*). Both types of relations can be extracted automatically from a corpus. Our approach focuses on synagmatic relations, in contrast to paradigmatic relations that appears to be the typical type of relation obtained by models such as LSA and word2vec (Rapp 2002; Peirsman, Heylen, and Geeraerts 2008).

Recently, the interest in multilingual metaphor processing has raised. Littlemore et al. (2018) took metaphors generated by *Metaphor Magnet*, translated them into Spanish and Mandarin Chinese, and asked participants to evaluate them from multiple criteria (e.g. meaningfulness, appreciation and speed in finding meaning) to find common metaphorical qualities across languages. Their research reveals that some metaphor qualities are indeed shared and correlated among multiple languages. Another research is conducted by Shutova et al. (2017) where the authors have experimented with semi-supervised and unsupervised methods to identify metaphorical expressions in a text. While the method concentrates on multilingual metaphor identification, it marks the initial steps in the path to multilingual metaphor interpretation.

2.3 Metaphor Interpretation in Generation

Using apt metaphors to express ideas and emotions illustrate creativity. Thus, researchers in the field of computational linguistic creativity have been studying various ways to generate novel metaphors computationally. Novel metaphors are new, at least to the context they are used in, and apt, while dead (or conventional) metaphors are metaphorical expressions that we use in our daily lives without regarding them metaphorical (e.g. “time is running out” and “deadline”).

There exist various efforts to produce metaphors and metaphorical expressions automatically e.g. using word associations and semantic relations (Galván et al. 2016; Hämäläinen 2018a; Veale and Li 2013; Xiao and

Blat 2013; Veale and Li 2012), deep-learning methods (Gargett, Mille, and Barnden 2015; Abe, Sakamoto, and Nakagawa 2006), and semantic models (Gagliano et al. 2016). We dedicate this section to methods that interpret metaphors as part of the generation process whether it is to generate metaphors or metaphorical expressions.

2.3.1 Generation of metaphors

Martin (1990) introduced a component for expanding an input metaphor as part of a system called *MIDAS*, which revolves around uses of conventional *UNIX* metaphors (e.g. "I am *in* Emacs" and "*killing* a process"). Given the conventional nature of the metaphors, these expressions are hardly considered metaphorical by a layman yet their initial usage that of a metaphor. The component expands metaphors by searching for existing metaphors that are related to it. The method then evaluates, as an interpretation phase, whether the found metaphors can substitute the existing one while retaining the meaning; if so, the new metaphor is created and saved. The interpretation process makes use of syntactic and lexical information of the input expression and examines if existing interpretations would satisfy the requirements imposed by input (e.g. the input "*get into* lisp" would be interpreted as "*enter* lisp").

Another example of a method where metaphors are expanded based on their interpretations is *Metaphor Magnet* (Veale and Li 2012). *Metaphor Magnet* considers the interpretations of a metaphor (the interpretations process is described earlier) to be the expanded metaphors if they are consistent with the metaphorical viewpoints of the input.

While some sort of reasoning or evaluation is necessary (e.g. ensuring that the vehicle is strongly associated with the property to be expressed by the system) to generate apt metaphors, existing methods for metaphor generation do not evaluate whether the produced metaphor is more likely to be interpreted as intended despite some having the ability to interpret and generate metaphors. This is probably due to using the same knowledge and resources in both processes, which renders applying the interpretation process to the generated metaphors futile.

It is crucial to assess (approximate) how the audience would comprehend the generated metaphor as doing so would improve the quality and aptness of the produced metaphor by making it possible for the machine to chose the best candidates among the set of alternatives. To estimate how others would perceive the generated metaphor, a method that predicts interpretations of the generated metaphor would be needed. Having the generation and interpretation models isolated would allow exploring a wider space of

possibilities but in the case of using an interpretation process to discover new metaphors (like the ones described above in this section), a different way to interpret the expanded metaphors would be needed to assess the meaning.

As part of Paper III, we research the idea of applying a metaphor interpretation model (*Meta4meaning*) to pick metaphors generated by a simple metaphor generation method that produces metaphors from word associations obtained from the knowledge bases shared by Veale and Li (2013) and Alnajjar et al. (2017). The goal of employing the metaphor interpretation model is to identify apt metaphors that are more likely to result in the desired meaning.

The metaphor generator of Paper III takes in a tenor and an adjectival property as input. It then queries the knowledge bases to retrieve nouns (vehicle candidates) that are strongly associated with the intended property. For each of the vehicle candidates, a nominal metaphor (i.e. in the form of "***T****enor* is [a/n] ***V****ehicle*") is constructed. These metaphors are then passed to the metaphor interpretation model *Meta4meaning* to examine whether 1) the intended property is a valid interpretation of the metaphor and 2) the intended property is ranked higher in the interpretations of the generated metaphor "T is [a/n] V" than its reverse metaphor "V is [a/n] T". The second criteria is enforced to counter for the asymmetrical nature of metaphors. The results show that interpreting the generated metaphors computationally outperformed the baseline of generating metaphors solely based on their strong associations in conveying the intended meaning.

2.3.2 Generation of metaphorical expressions

There are numerous methods for producing figurative expressions, as I will describe in the following chapter but these methods take the interpretations of the expressions they produce for granted. In Paper III and V, we introduce aesthetic functions to measure the metaphoricity and the metaphorical meaning of expressions. These two functions are described below.

Both functions take three parameters as input, an expression $\mathcal{E}$, tenor T and vehicle v, and use a semantic relatedness model such as the one build as part of Paper I, *Meta4meaning*. The purpose of the first function, $f_{\text{metaph-maxrel}}$, is to measure the semantic relatedness of words in the expression to the tenor and vehicle. To do so, it considers the strongest relationships between any of the content words t in the expression $\mathcal{E}$, and the tenor T and the vehicle V:

$$maxrel(\mathcal{E}, w) = \max_{t \in \mathcal{E}} \omega(t, w) \tag{2.1a}$$

$$f_{\text{metaph-maxrel}}(\mathcal{E}, T, V) = maxrel(\mathcal{E}, T) \cdot maxrel(\mathcal{E}, V), \tag{2.1b}$$

where $\omega(\cdot)$ is the semantic relatedness score returned by the model. A positive value returned by $f_{\text{metaph-maxrel}}$ indicates that the expression contains a word that is related to the tenor and another (possibly the same) word that is related to the vehicle. The bigger the value, the stronger the relatedness of these words to the tenor and vehicle. This function is introduced to ensure that the generated expression has words related to the tenor T and vehicle V, as it can hardly be metaphorical in the intended manner if it did not include words (strongly) related to both concepts.

The second metaphoricity function, $f_{\text{metaph-diffrel}}$, looks for a word t in the expression that is related to the metaphorical vehicle V but *not* to the tenor T. The hypothesis is that such a word t is more likely to force a metaphorical interpretation of the expression, in order to connect t to the tenor T. The function is defined as follows:

$$f_{\text{metaph-diffrel}}(\mathcal{E}, T, v) = \max_{t \in \mathcal{E}} (\omega(t, v) - \omega(t, T)). \tag{2.2}$$

To illustrate how these two functions work, let the input tenor T be *car* and the (input or automatically generated) vehicle V be *dancer*. Imagine the NLG system is working with the expression "The cars of ***_NOUN" where it is in the phase of picking a suitable noun that makes the expression metaphorical. To narrow this example, assume that the system needs to pick this noun out of three candidate words, *driver*, *street* and *stage*. As the expression already contains a word that is related to the tenor, i.e. *cars*, the first metaphoricity function will guide finding words related to vehicle. Accordingly, the candidate *driver* will receive the least score as using it yield an expression carrying no relatedness to the metaphorical vehicle. While *street* relates to both concepts, the tenor and the vehicle (e.g. *street dancer*), its relatedness is stronger to the tenor. The candidate *stage* is solely related to the vehicle.

The second metaphoricity function comes in to pick the candidate that is more likely to have a metaphorical interpretation by encouraging using a word that is related to the vehicle V but *not* to the tenor T. Following our previous example, the function would assign a higher score to the candidate *stage* than *street*, since the word *stage* is *not* related to cars. Thus, the system would generate "The cars of stage." as it is the most metaphorical option.

Now, we will look into how these two functions were utilised in Papers III and V to produce metaphorical expressions, English slogans and Finnish poems, respectively. In Paper III, the two metaphoricity functions are combined (summed) together and given equal weights (importance) to represent the internal dimension of the genetic algorithm concerning metaphoricity that are used to guide the generative process. The method contained three other internal dimensions for assessing relatedness, grammaticality and prosody.

The two metaphoricity functions were used in Paper V to generate Finnish poetry. The proposed approach consisted of two types of systems, a *master* that is a genetic algorithm and an *apprentice* which is a sequence-to-sequence neural model. The metaphor interpretation and metaphoricity functions were utilised in the *master*. In short, words in the generated poem were clustered based on their semantic distance to each other. Each cluster were then represented by a single word, the centroid. The method then iterates over all the possible combinations for having two clusters as tenor-vehicle and measures the metaphoricity score in a similar fashion to Paper III. From the returned metaphoricity scores, two aesthetics were defined: 1) the maximum metaphoricity score and 2) the number of metaphorical word clusters (i.e. tenor-vehicle pairs with a positive metaphoricity score).

Based on the conducted evaluations in both papers, these functions appear to estimate and increase metaphorical interpretationality of generated short expressions (e.g. slogans) and text (e.g. poems). Furthermore, our work in Paper V demonstrates the transferability of these functions to languages other than English as they were applied successfully to measure metaphoricity in Finnish.

Even though these functions aid in producing metaphorical expressions, they cannot guarantee that the expressions convey the desired meaning. In Paper III, a different internal dimension of the genetic algorithm would optimise the relatedness of the expression to the tenor and the desired property to encourage using words that would be associated with the intended meaning. Despite this dimension, the meaning of the entire expression is unclear to the method. To reduce such uncertainty, a subsequent process for interpreting the generated text as a whole would be required, which is a great future step to enhancing the current state.

Chapter 3

Generation of Figurative Language

Figurative language generation is a subdomain of creative natural language generation. While figurative language is an important characteristic of many linguistic artefacts such as songs, slogans and poetry, it is not always tackled explicitly in the approaches to generate such artefacts. For instance, there is a growing body of recent work on poem generation that leaves figurative language of the output to a mere chance (Yi et al. 2018; Yang et al. 2019; Hämäläinen and Alnajjar 2019b; Van de Cruys 2020).

In these approaches, the generative model itself is not at all aware of the phenomenon of figure in language, but rather produces expressions of that kind accidentally, so that people can read more into the output and interpret it in the light of it having figurative language. In fact, Gervás (2017) argues that many poem generation systems pick only a specific feature of poetry to be modelled. This is done implicitly by usually focusing on simple features such as rhyme, meter and semantic cohesion, however as this narrow focus is hardly ever stated, the reader of these papers may be lead to believe more complex problem of poem generation was solved.

In this chapter, I focus on the existing work on generation of figurative language; especially generation of colour names, slogans, humour and poems, given the wide range of natural text types that contain figurative language. Moreover, I present my own work conducted in Papers II-V. It becomes evident in Papers III-V that figurative language is to be explicitly modelled and its emergence in the final output cannot left to a mere chance. In Papers IV and V, the need of figurative language is explicitly stated in the theoretical definition of the problem.

3.1 Naming Colours

Colour is a very peculiar notion and surprisingly tricky for a computational system to grasp in a human like fashion. For colours are not a phenomenon of the physical world, but are a creation of the mind, conscious experiences, also know as qualia. Qualia are indeed a hard problem to solve even in the philosophy of mind (Chalmers 1995).

As qualia refer to conscious experiences, it becomes difficult for something non-conscious as a computer to deal with them in a human like fashion. Furthermore, even though there is an agreement to a degree between different people on what colour is shown when they are exposed to a stimulus, there is still individual variation in how the cone cells in the eyes react to lights of different wavelengths (Nerger, Volbrecht, and Ayde 1995). This results in differences in how colours are perceived already in the level of the raw sensory data.

There are also cultural differences in how colours are represented. According to the linguistic hypothesis of relativity (see Kay and Kempton (1984)), one might think that this means that people will perceive colours inherently differently in different cultures, however, the categories of colours in different languages are not very different (Lindsey and Brown 2006). Humans associate different emotions and concepts to colours (Odbert, Karwoski, and Eckerson 1942; D'Andrade and Egan 1974) in different cultures. For instance, the colour "blue" is associated with sadness in the English speaking world, while other cultures such as the Finnish or the Arabic one do not have any particular emotional association with that colour. Culture specific notions also have their typical colour associations such as "red" during the end of the year with Christmas in the western culture and "yellow" with Eid al-Fitr in the Arabian culture (due to yellow being a typical colour of the crescent and fanous).

We all have a common colour representation for the words "red", "black" or "blue", despite the slight personal variation. But when referring to a certain shade of a colour, we resort to objects that typically manifest the shade to describe it. For instance, take a look at the different shades of the colour blue shown in Figure 3.1. What would be an apt name to describe and distinguish them? By contemplating these colours and stereotypical associations with them, possible names such as "sky", "sea" and "cobalt" emerge, respectively.

Coming up with apt and novel names and usages of names exhibit creativity, like Walt Whitman's usage of the phrase "boundless blue" to refer to the *sea* in "Chant on, sail on, bear o'er the boundless blue from me to every sea" (Whitman 1855) and Homer's epithet "*wine-dark* sea", in the

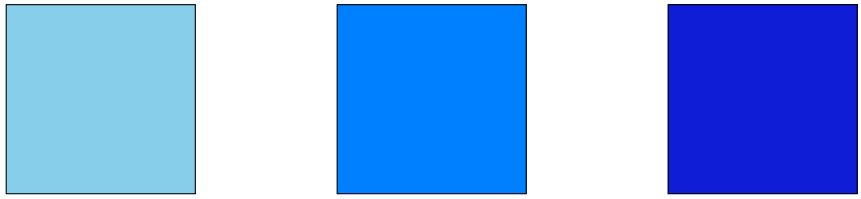

Figure 3.1: Three different shades of the colour blue.

Iliad and Odyssey, to paint a mental image of a stormy sea. In the light of these examples, we can, indeed, gather that colours have their use in creative language.

The interest in researching methods concerning colours and names has been rising lately, not only because it has multiple applications as stated in Heer and Stone (2012) but also because it helps us to understand how we parse the world around us, linking language to our conceptual-intentional system through heuristical notions like colour. XKCD (Munroe 2010) has conducted a survey asking volunteers to name/describe a given RGB code. The results of the survey are publicly available, which opened the door for researchers to study and model associations between the perception of colours and objects, and the language we use to describe or name colours. During the survey, users were able too answer with straightforward or creative names, and the length of the answers varied from a single word to a whole sentence or paragraph. Examples of names in the dataset are "yellow", "pistacio", "mustard yellow" and "ultra ripe banana".

Havasi, Speer, and Holmgren (2010) build a system that colours words in a text. Their system uses the XKCD dataset along with a Python library called NodeBox and ConceptNet. NodeBox[1] contained a small set of manual mapping of concepts to their typical colour associations (e.g. *Christmas* and red and green). ConceptNet (Speer, Chin, and Havasi 2017) is a network of concepts connected with labelled edges (e.g. relations) such as *HasProperty* and *PartOf*) built by crowdsourcing. Concepts appearing at least three times in the colour names in the XKCD dataset are mapped to the corresponding colours values, which are then normalised because very similar shades of a colour could be distant in the RGB colour space. For concepts that are neither in the XKCD dataset nor in the NodeBox, the system predicts a colour based on its semantically similar words that have a colour.

The work of Setlur and Stone (2016) addressed the same problem of colouring terms in a text. They mined words associated with basic colours

[1]https://github.com/nodebox/nodebox

(e.g. blue, red and black) from XKCD's dataset and Google n-gram (Brants and Franz 2006). A pointwise mutual information score is measured from co-occurrence frequencies between basic colours and words associated with them. To acquire colour mappings for words, their method expands the mined word associations using WordNet (Christiane and Brown 2005) and, then, it sends a query to Google Image Search to find images related to the words. A predominant colour is deduced for each word by clustering the colours in the images using k-means clustering.

Heer and Stone (2012) used the XKCD dataset to construct a probabilistic model that maps between colours and names from a colour-name count matrix. From the model, an estimate of how uniquely a colour is named and a distance measure based on colour names can be calculated. Neural network approaches were presented by Kawakami et al. (2016) and Monroe et al. (2017). Kawakami et al. (2016) trained a neural network model to predict colours for a sequence of characters, acting as names or descriptions, collected from COLOURlovers[2]. They then tested their model on the XKCD dataset. Monroe et al. (2017) suggested an approach consisting of multiple neural networks to interpret colour descriptions in a context of three colours.

Majority of the previous work concentrated on mapping concepts to colours. While we tackle the same task as part of Paper II, the focus of our work is rather the reverse, i.e. producing a creative/descriptive name for an input colour. Moreover, our approach takes a different path in acquiring associations with colours and naming them. Our method is based on the linguistic readymade (Veale 2012) that is inspired from Marcel Duchamp's idea in art where something (a phrase in our case) is taken from its conventional context of use and used in a new context that gives it new meaning and new relevance. We utilise a technique called Creative Information Retrieval (CIR) (Veale 2011) to obtain stereotypical associations of nouns with basic colours and linguistic readymades that are used as names from Google n-grams (Brants and Franz 2006). CIR defines operators such as @*Adj* and ^*Noun* for retrieving stereotypical associations with the adjective *Adj* (e.g. @*hot* would match *summer* and *lava*) and words that are nouns in a text (e.g. "a blue ^*Noun*" would match *sky* and *sea*), respectively. We retrieved stereotypical associations with 11 basic colours by retrieving all nouns matching the CIR query "^*Noun* - *colour*" (e.g. *cherry-red*) in Google 3-grams, where colour is the basic colour such as red, green and blue. For *colour* = *red*, the query yields nouns such as *cherry*, *blood* and *rose*. To retain high-quality and valuable associations for the task, we manually cleaned

[2]A creative community for sharing colours: `https://www.colourlovers.com/`.

the results to remove unwanted associations such as *lemon* and *tallahassee* with the colour red. RGB codes are then hand-assigned to these associations with respect to the basic colour (e.g. #E53134 to tandoori-red and #FD5E53 to sunset-red). This knowledge base of stereotypical associations and colour maps is publicly available[3].

When a user passes a colour (in RGB or hexcode format) to the method, the method uses the two analogous colours it has in the colour wheel and converts them to CIE LAB code (Sharma and Bala 2017, 29–32) as the space aims to model human visual perception. All stereotypical associations retrieved in the previous phase are also projected to the same colour space. The method then highlights the closest stereotypical associations to each of the analogous colours using the Delta E CIE76 distance function that are within the empirically set threshold 14. Next, it seeks readymades in Google uni/bi-grams containing matched stereotypes of both analogous colours, which are then treated as candidate colour names. For example, let #FCF9F0 be the input colour which has #FCF3F0 and #F9FCF0 as its analogous colours. *seashell*-white (#FFF5EE) and *pearl*-white (#F7FBEF) are the closest stereotypes to these analogous colours, in the same order, which also appear in Google bi-grams (e.g. "seashell pearl"). To further produce names fitting a desired theme, an optional categorization of stereotypical associations can be imposed as conducted in the paper.

Even though the evaluation results of our system far exceeded those of human-written colour names on COLOURlovers, the creativity of the system is much in the eye of the beholder. As Ritchie (2007) argues, this is enough for a system to be creative; if the output is accepted as creative by people, the process that produced the artefacts is irrelevant. However, taking this approach to computational creativity can take the field only so far. As we will see in the following sections, it is important to model creativity on a much deeper level than that of the mere output.

3.2 Generation of Figurative Language

A great focus of the computational creativity field is the production of creative artefacts. Language is a common medium for conveying ideas and messages creatively by utilising figurative language properly. In this section we describe different takes on generating figurative language, mainly slogans, humour and poems, from a technical perspective, leaving the evaluation of their creativity to the next chapter. We begin with a brief introduction of the related work and then present our methods.

[3] `https://www.cs.helsinki.fi/en/node/80968`

3.2.1 Generation of Slogans

Slogans are short expressions that are used in advertising campaigns to increase awareness and recall of the brand while distinguishing it from competitors. In Paper III, we define a slogan, from an advertising perspective, as a concise, advertisable, and autonomous phrase that expresses a concept (e.g., an idea, product, or entity); the phrase will be frequently repeated and associated with the concept. Figurative language is commonly found in slogans (Miller and Toman 2016) because of their positive effect on increasing persuasiveness, catchiness and memorability (Reinsch Jr. 1971; Reece, Van den Bergh, and Li 1994; Tom and Eves 1999), which makes them interesting to research in the field of computationoal creativity.

Inspired by the the "optimal innovation hypothesis" (Giora 2003) which indicates that optimal innovation is a result of accompanying novelty with familiarity, Strapparava, Valitutti, and Stock (2007) have proposed a method that modifies an input expression by replacing word in it with candidates having semantic and emotional relatedness to a desired topic, and assonance. *BrainSup*, by Özbal, Pighin, and Strapparava (2013), generates expressions by filling a syntactic parse tree (skeleton) of existing expressions with words meeting certain criteria such emotional effect, domain relatedness, syntactic constraints and phonetic properties. The replacement words are found using beam search, a greedy search. Another approach is *Figure8* (Harmon 2015), which generates metaphorical short expressions by accepting a tenor as input and then filling in manually-crafted templates of metaphorical and simile expressions with words passing predefined criteria (clarity, novelty, aptness, unpredictability and prosody).

Tomašič, Papa, and Žnidaršič (2015) have employed a genetic algorithm approach to generate slogans. The approach starts by automatically extracting keywords from descriptions of the brand/product. Slogan skeletons are then automatically filled and evolved with the genetic algorithm while optimising a single dimension constituted of multiple functions such as expression length, word frequencies and semantic relatedness. *BISLON* (Repar et al. 2018) applies the idea of cross-context associations, "bisociations" (Koestler 1964), to produce slogans. The method does so by taking in documents or terms related to the desired target concept and the bisociated one, which is followed by highlighting keywords for both concepts and expanding them using word embeddings. Next, the method fills skeletons that are built from existing slogans with terms from both sets of keywords based on their part-of-speech, while considering prosody features (alliteration, assonance, consonance, and rhyme). Finally, the method recommends slogans that have a high semantic similarity score to the input

and semantic cohesion estimated by a language model.

In Paper III we present a method for generating slogans. The method takes in a concept, representing the product or the brand, and an adjectival property defining the desired message to convey. The method consists of two components, a metaphor generator based on a metaphor interpretation model (c.f. Section 2.3.1) and a slogan generator based on a genetic algorithm. For the input concept and property, the metaphor generation component produces a list of apt metaphorical vehicles that highlight the desired property. This component is introduced to allow generating metaphorical expressions.

The next step in the process, takes in a vehicle from the produced list and builds semantic spaces containing words related to the three concepts, i.e. concept, property and vehicle. Using existing slogans as skeletons, an initial population of slogans is created and filled with words fitting the syntactic dependencies. The genetic algorithm modifies the initial population by mutating and crossing over the slogans for a number of generations while optimising four main criteria. These criteria are 1) relatedness to the input, 2) language correctness, 3) metaphoricity and 4) prosody. Each criteria is a combination of multiple functions, e.g. metaphoricity criteria is an average of the relatedness to the concept and vehicle (Equation 2.1), and relatedness to the vehicle but not the concept (Equation 2.2). For the optimisation function, a non-dominant sorting algorithm (NSGA-II; Deb et al. (2002)) is used given its ability to perform multi-dimensional optimisations.

Our results show that the method is capable of producing both successful and metaphorical slogans. Furthermore, the results indicate that slogans with balanced criteria were considered better overall in comparison to maximising individual criteria; hence, optimising multiple dimensions is recommended when producing creative artefacts. While these results are positive, the method is intended as an auxiliary tool aiding advertising professional explore potential slogans and should not be used in production. This is to prevent any unintended repercussions (e.g. accidental generation of offensive language) and because the quality of an average slogan produced by our method, and any of the existing methods, is far from hand-crafted slogans by professionals.

Comparing our method to the relevant methods described above, we can highlight a couple of important differences. The first difference is that our method focuses on generating advertising slogans for a product that highlight the desired adjectival property defined by the user, while the rest solely consider the concept when generating figurative expressions. Secondly, a metaphor is used to express the property indirectly. This metaphor is auto-

matically generated to cater for the requirements set by the task. While it may oftentimes be the case that the methods above output expressions of a metaphoric quality, less control is given to the system in terms of the meaning conveyed by the metaphor. An exception to this is BISLON (Repar et al. 2018), where a bisociated concept is input to the system which has the potential of being the vehicle of the metaphor. Additionally, in our original paper we examine several internal evaluation functions used by our method, in order to gain insight into their value in generation of metaphorical slogans.

3.2.2 Generation of Humour

Generation of humour has received interest at least for a decade (Ritchie 2005; Hong and Ong 2009; Valitutti et al. 2013; Costa, Gonçalo Oliveira, and Pinto 2015). We dedicate this section to describing the main theories and some of the most recent work conducted on the topic.

Humour is an inherent part of being a human and as such it has provoked the interest of many researchers in the past to formulate a definition for it (see Krikmann 2006). Koestler (1964) sees humour as a part of creativity together with discovery and art. In his view, what is characteristic to humour in comparison to the other two constituents of creativity, discovery and art, is that its emotional mood is aggressive in its nature. He calls bisociation in humour the collision of two frames of reference in a comic way.

Raskin (1985) presents a theory that is not too far away from the previously described one in the sense that in order for text to be humorous, it has to be compatible with two different scripts. The different scripts have to be somehow in opposition, for example in the sense that one script is a real situation and the other is not real.

In Attardo and Raskin (1991) humour is seen to consist of six hierarchical knowledge resources: language, narrative strategy, target, situation, logical mechanism and script opposition. As in the previous theories, the incongruity of two possible interpretations is seen as an important aspect for humour. An interesting notion that we will take into a closer examination is that of target. According to the authors it is not uncommon for a joke to have a target, such as an important political person or an ethnic group, to be made fun of.

Two requirements have been suggested in the past as components of humour in jokes: surprise and coherence (see Brownell et al. 1983). A joke will then consist of a surprising element that will need to be coherent in the context of the joke. This is similar to having two incongruous scripts being

simultaneously possible.

Veale (2004) points out that the theories of Raskin (1985) and Attardo and Raskin (1991) entail people to be forced into resolution of humour. He argues that humour should not be seen as resolution of incompatible scripts, but rather as a collaboration, where the listener willingly accepts the humorous interpretation of the joke. Moreover, he argues that while incongruity contributes to humour, it does not alone constitute it.

Pun generation with a neural language model is one of the most recent efforts on humour generation (Yu, Tan, and Wan 2018). Their approach consists of training a conditional language model and using a beam search to find sentences that can support two polysemous meanings for a given word. In addition they train a model to highlight the different meanings of the word in the sentence. Unfortunately, they evaluate their system on human evaluators based on three quantitative metrics: fluency, accuracy and readability, none of which tells anything about how funny or apt the puns were.

Surprise is also one of the key aspects of a recent pun generator (He, Peng, and Liang 2019). They model surprise as conditional probabilities. They introduce a local surprise model to assess the surprise in the immediate context of the pun word and a global surprise to assess the surprise in the context of the whole text. Their approach retrieves text from a corpus based on an original word - pun word pair. They do a word replacement for local surprise and insert a topic word for global surprise.

Valitutti et al. (2016) have proposed a method for turning a given text into a humours one by substituting a single word in it while considering constraints such as similarities between the replacement word and its substitution and semantic constraints to introduce a taboo word while fitting the context. They have conducted an empirical evaluation and found that the usage of taboo words had a positive effect on the humour.

An approach building on humour theories is that of Winters, Nys, and De Schreye (2019). The theories are used in feature engineering. They learn templates and metrical schemata from jokes rated by people with a star rating. They embrace more traditional machine learning techniques over neural networks, which has the advantage of a greater interpretability of the models.

Humour has also been tried to recognise automatically in the past. One of such attempts focuses on extracting humour anchors, i.e. words that can make text humorous, automatically (Yang et al. 2015). A similar humour anchor based approach is also embraced by Cattle and Ma (2018). Both of the approaches rely on feature engineering basing on humour theories.

Recently LSTM models have been used for the task of humour detection with different rates of success (Cai, Li, and Wan 2018; Sane et al. 2019; Zou and Lu 2019).

In Paper IV, we present the so-called master-apprentice framework where the master is a genetic algorithm based on our earlier method for generating slogans, and the apprentice is a neural network implemented in OpenNMT (Klein et al. 2017). We used this framework to generate humorous and satirical movie titles out of existing ones. The main difference between the implementation of the genetic algorithm in Paper III and the one representing the master is that it works on a word-level replacement and optimises different aesthetics (except prosody) for optimising humour.

The main idea behind the master-apprentice framework is to have an interpretable system, the creativity of which can be motivated, and a neural black-box model that can learn to create outside of the scope we have defined. A genetic algorithm is a suitable master as we can define by ourselves how the initial population is formed, how the genetic process takes place and what aesthetics are employed at the time of selecting the fittest creative artefacts. This type of a system is very interpretable as we can debug and see every single step taken by the system, in other words, we can know why certain output got produced. However, this type of a master cannot go beyond what we have defined: it will follow the same aesthetics whenever it picks the fittest individuals. The role of the apprentice is to go beyond this, as it can learn from the master and human authored data, it can explore a very different set of possible solutions than what the master is capable of. In other words, it is approximating creative autonomy, as the changes to its standards are not random, but emerge from the training data and can be altered by fine tuning. However, the creativity of an apprentice alone is a harder thing to motivate as the reasons why it ends up producing certain output are less clear. Furthermore, a neural network model is bound to repeat features from its training data, where as a genetic algorithm can come up with something novel as it does not rely on training on human authored data. Therefore both parties, the master and the apprentice contribute to the overall creativity to the system.

The optimisation metric aims for low semantic similarity of the replacement word with the original word to maximise surprise and high similarity with the satirical target to maximise coherence. Punyness is modelled in the method through the prosody fitness functions of the genetic algorithm, but the output is not strictly limited to puns.

For our case, the apprentice is an recurrent neural network (RNN) model that continuously learns from expressions generated by the genetic algo-

rithm (the master) and real people. This allows the method to create its own standard of what is punny and suitable, and approximate creative autonomy by changing its standards while continuously learning from real people.

3.2.3 Generation of Poems

Poems are a great medium for expressing ideas and emotions by using creative figurative language. Various researchers have worked on automatic approaches for producing poetry (Gervás 2001; Lamb and Brown 2019; Misztal and Indurkhya 2014; Gonçalo Oliveira et al. 2017), from rule-based systems to end-to-end neural networks (Yi et al. 2018; Li et al. 2018; Yang et al. 2018). Our approach in Paper V generated Finnish poetry. Finnish is a morphologically rich language, which adds an additional challenge for NLG systems such as a poem generator. In this section, we shortly describe the technicalities of the most relevant work followed by an overview of our method presented in Paper V. For more details regarding related work, Gonçalo Oliveira (2017) has conducted a thorough survey on the topic of automatic poetry generation.

TwitSong (Lamb and Brown 2019) aligns verses that rhyme together and computes a score for each verse on four criteria, which are 1) meter, 2) emotion, 3) topicality and 4) imagery. A genetic algorithm approach is employed in TwitSong to alter verses with a low score.

Toivanen et al. (2012) proposed an approach for generating Finnish poetry. The approach takes in a target concept as input. It then retrieves words related to the input from a background corpus and uses these words to replace words in an existing poem while satisfying syntax and morphological constraints. Hämäläinen (2018a) also dealt with generating Finnish poems. His method uses a repository of syntactic dependencies that are used to fill hand-crafted verse templates.

The approach of Colton, Goodwin, and Veale (2012) is template-based and it produces poems tailored to a news article. While their approach does not deal with Finnish, it follows the FACE model (c.f. Section 1.1.2), which is the same model used to assess the creativity of our method.

Our method extends the work conducted in Paper IV and the work presented by Hämäläinen and Alnajjar (2019b). From a technical perspective, there are four major differences to the earlier approach, which are 1) instead of dealing with short expressions, the method is tested our to produce stanzas ($\approx$ 5 verses), 2) the method produces poems in Finnish, 3) aesthetics functions are tailored for poem generation and their weights are automatically learned from a poem corpus, and 4) the master is capable of supplying

its opinion (master's liking) of poems to the apprentice.

In contrast to a majority of the recent existing work on poem generation, our method does not make any implicit claims on solving poem generation as a whole while mainly focusing on the superficial such as rhyme, meter and simple semantic cohesion. In contrast, out work tries to capture deeper features of poetry such as metaphor, semantic cohesion, sentiment and concreteness. These notions come from the literature on poetry and they are explicitly modelled in the fitness functions of the master.

Surface realisation is a difficult problem when it comes to Finnish. While in English, words tend to appear in their dictionary form in a sentence, in Finnish they mostly need to be inflected in one of the cases according to morphosyntactic rules. Some of these rules can be resolved by inflecting the new substitute words with the same morphology as the original word in the poem. For this we use Omorfi (Pirinen et al. 2017) through UralicNLP (Hämäläinen 2019). However, this only accounts for the morphosyntatic phenomenon known as agreement. If a verb gets changed in the sentence, its case government rule might be different from the original verb. In this case, we apply Syntax Maker (Hämäläinen and Rueter 2018) to resolve the case of the complements of the new verb. Taking these two different linguistic rules into account while generating results in more syntactic output. The surface realisation is conducted always before using the fitness functions to ensure that functions such as rhyming look at the actual rhymes, not at the rhymes of the lemma.

Chapter 4

Evaluation of Creative Systems and Expressions

Evaluation in the field of computational creativity is not an easy problem to be solved. However, it is a fundamentally important one, because it is one of the few ways of measuring progress. Surely enough it is easy to come up with yet another system that generates humour or poems. But without any good evaluation methods it becomes impossible to say whether any progress was made by introducing a new, more complex, system to solve the problem.

While automated evaluation metrics can work for close-ended tasks, such as F-scores and accuracy for tasks such as parsing and tagging or BLEU score for machine translation, they still have their shortcomings. For instance, BLEU should only be used for development time debugging rather than to prove scientific progress (Reiter 2018), and systems scoring high on one dataset drop their performance drastically when tested on different benchmarks (Talman and Chatzikyriakidis 2019). Nevertheless these evaluation methods make it possible to measure progress in a more objective fashion.

However, due to the nature of computational creativity, it is difficult to evaluate the output entirely computationally because there is no limited set of possible solutions to compare against. And in fact, we have found (Hämäläinen and Alnajjar 2019a) that, in paraphrase generation, simple metrics such as BLEU and PINC scores that have been used in evaluating text paraphrasing (Tiedemann and Scherrer 2019) simply are not sufficient, as they are very poor in predicting the results of a human evaluation.

4.1 Evaluation for the Sake of it

Ritchie (2007) states that the creativity of a computational system can be determined by the artefacts that it produces, if it exhibited *novelty*, *quality* and *typicality*. Most of the evaluation conducted for computational systems producing creative artefacts focus solely on the output. While it is crucial for a creative system to produce creative results, using only the output as a measurement for creativity is not sufficient as we will show in this section.

In fact, we have found that, when the evaluation does not correspond to the features that have been modelled as no definition of creativity or framework (e.g. creative tripod and FACE) has been used to back up what has been modelled, even superficial non-creative features such as dialect can make the results appear more creative and original according to human judges (Hämäläinen et al. 2020).

In Paper II, we propose a method for naming colours. We evaluated the method by running a crowd-sourcing experiment in which we showed judges a total of 2587 colours and for each of them two, randomly ordered, names. One name was generated by our method and the other was a human-written name acquired from `COLOURlovers.com`. For each colour, three-to-five judges were asked to answer four questions, 1) which name is more descriptive, 2) which name do you prefer, 3) which name seems the most creative for the colour shown and, for qualitative analysis, 4) why did you answer these questions the way you did.

Overall, 70% of the answers were in favour for names generated by our method on the first three questions. These results clearly indicate that judges found the generated names are more descriptive and creative. Based on the view of Ritchie (2007), our method is creative. While the output exhibits creativity, I personally would attribute the creativity to the creators of the method, not the method itself.

This is mainly due to the algorithmic design of the method and the fact that the method relies heavily on manual assigning of colour codes and filtering of mined associations. Although such parts that are done by hand could be automated as done in (Havasi, Speer, and Holmgren 2010), the process of using the high-quality knowledge to find suitable names is fully defined by the authors; which limits the freedom of the method.

In the following sections, we describe two types of evaluating the creativity of the method that consider more than just the output, as done in Papers III-V. The first evaluation type focuses on the features that are being modelled in the creative system. The second evaluation type additionally demands the computer to provide an explanation (i.e. framing) for

the creative artefact.

4.2 Evaluating the Features Modelled

In Paper III, we tested our slogan generator by running a crowdsourced evaluation on `figure-eight.com`. The questions used in the evaluation were based on the four dimensions that were modelled in our method, which are 1) relatedness to the input (concept and property), 2) language correctness, 3) metaphoricity and 4) prosody. Additionally, we asked a fifth question to examine the overall suitability of the produced expression to be used as a slogan.

The results of the evaluation showed that the method is capable of producing good slogans. More importantly, they showed us whether the modelled dimensions had an effect on the quality of the generated slogans. For instance, the language correctness dimension did not improve the grammaticality of the slogans which is probably because, by design, actions such as filling of a skeleton and mutating an individual slogan took grammaticality into account by ensuring that the syntactic dependencies are satisfied. However, the relatedness dimension appeared to contribute the most to the quality of the slogans. A final remark from the results is that combining and balancing these internal dimensions produced the best slogans overall.

Despite manually defining the internal dimensions of the method, the method is free to investigate different paths as a part of the genetic process. This means that the method is stochastic, i.e. running the method multiple times for a single input would yield different results depending on the random process that guides its path. As with any genetic algorithms approach, certain hyperparameters need to be defined (e.g. population size and number of iterations). In our case, we empirically set them but a meta-layer of hyperparameter optimisation could be utilised to free the method further from any programmer-defined restrictions.

In contrast to Paper III, in Paper IV, we embraced a more theoretical approach both in creativity and in humour. Based on our definition following the creative tripod (Colton 2008), we model the different parts contributing to the creativity of the system. These parts are explicitly modelled in the workings of the master; Imagination is modelled by the genetic approach that produces new artefacts without a dataset to learn from. Skill is modelled in the type of input the system takes and what the mutation and crossover functions do during the genetic process. And finally, appreciation is modelled in the fitness functions of the system.

The evaluation questions used in the human evaluation followed directly

from the initial problem definition and what was modelled in the master in the spirit of the guidelines for evaluation of computational creativity set by Jordanous (2012). This is important since we want to try to avoid the possibility of people interpreting too much more into the puns than what the systems was aware of during the creative process. In other words, in this evaluation we only evaluate what was being modelled in the master.

This evaluation, however, is trickier for the apprentice that ended up scoring higher than the master. Nothing was really specifically modelled in the apprentice; it just happened to learn some type of puniness from the data. This evaluation is not to the point in the same fashion as that of the master. And quite often, this remains the problem for the latest neural models that are proposed to solve creative or generative tasks. The degree to which the evaluation scores are attributable to the model itself rather than to the combination of the training data and people's willingness to interpret the generated artefacts is hardly ever discussed in the literature.

In our follow-up paper (Hämäläinen and Alnajjar 2019c), we experimented with a multitude of different ways of training an apprentice to get a better grasp on how the training influences the output. However, our results in the same evaluation metrics ended up being hard to interpret. No single method scored the highest on all the metrics and in the end their intercomparison turned out to be difficult. In the next section, I will shed some light into how we improved the evaluation method even further by more directly exposing the internal aesthetics of the system for human evaluation.

4.3 Exposing the Internals for Evaluation

In Paper V, we continued the theory driven approach established in Paper IV. That is that creativity should be first defined, then implemented accordingly, and naturally the evaluation questions would follow from what was implemented. This time, however, we changed the theory used to define creativity on an abstract-level. Instead of following the creative tripod, we used the FACE (Colton, Charnley, and Pease 2011) theory to define creativity.

The FACE theory introduces a notion of framing, something that became the most important part of the evaluation we conducted. Framing allows the computer to produce some explanation for its art. Instead of using this for persuasive effect, we formulated the evaluation through framing statements. The computer fills in template statements about the poem, and we measure people's agreement with the statements. The important part is that these statements are produced by the very same fitness functions as

the system uses while generating poetry. In this evaluation, we therefore assess the degree to which the requirements set for poetry initially were met by the aesthetics of the systems.

What we found especially helpful in this type of evaluation was to give people the possibility to indicate that they did not know whether to agree with not instead of forcing them to either agree or disagree. This revealed that people struggled in formulating an opinion on the most abstract-level questions, namely metaphors and semantic clusters. These results are interesting in the sense that if people find it difficult to say yes or no when presented with a metaphor and its interpretation, how can we assume that they can answer to more abstract questions on the generated poems. Like the typical evaluation questions used in the field.

Another interesting finding was in the evaluation results of the different types of rhymes. Even though our rule-based system can detected rhyme correctly[1], people would still not show a 100% agreement with these statements. In fact, it turned out that the mere presence of a rhyme was not enough to make it perceivable. Therefore, in the future, one should put more attention into the quality of the rhyming elements as well. Something that is barely discussed in the existing work.

The fitness function for sentiment analysis was based on a state-of-the-art sentiment analysis model (Feng and Wan 2019). However, based on our evaluation, it failed miserably. It only scored high in predicting negative sentiment but was very poor at predicting positive sentiment. This finding is important because it shows that sentiment analysis, even if it received convincing state-of-the-art results, is still far from solved in more difficult domains such as poetry, where sentiment is often conveyed indirectly through figurative language.

All in all, we find this particular evaluation method more revealing of the shortcomings of the system than the evaluation with abstract questions we conducted in Hämäläinen and Alnajjar (2019b). Mainly because this time around the evaluation was tailored to evaluate the exact analysis produced by the fitness functions. This means that any fitness function scoring poorly in the evaluation is an obvious place to start on improving the system in the future.

[1]The code has been released and is available for inspection on `https://github.com/mikahama/finmeter`

Chapter 5

Conclusion

A key part of the thesis is the metaphor interpretation method that is extended further to produce nominal metaphors and metaphorical expressions, and estimate the metaphoricity of expressions. The method has also been used successfully in Finnish in addition to English, which it was first developed for. Furthermore, we have presented various natural language generation methods for producing creative language, from naming colours to generating puns, slogans and poems.

The natural language generation methods we have presented in this thesis have performed well in the tasks they were designed to model. This is not only reflected in the human evaluation results of the output but also in the performance of the different aesthetic functions modelled. Although, comparing these methods with existing ones in terms of performance is difficult because of various reasons such as lack of standardised ways of conducting the evaluation, clarity in establishing the problem setting and narrowing down the context in which the problem is addressed, and availability of data and code.

We have also shown the importance of evaluating the creativity of the system. This can be achieved by defining what's meant by creativity so that when the system is modelled accordingly to the definition, its creativity can be assessed based on the definition.

In the context of our evaluation of generated slogans, our findings suggest that balancing multiple aesthetic dimensions (such as semantic relatedness and metaphoricity) outperformed maximising a single dimension. This also helped in identifying weak aesthetics that do not contribute positively to the final artefact. This was made possible by having the evaluation assess the individual aesthetics while still evaluating the final output.

From our experience from employing two distinct frameworks for creativity (namely creativity tripod and FACE), we can say that different

frameworks will highlight different aspects of creativity in the method. For example, the creative tripod framework highlights a requirement for novelty as imagination is an integral part of the theory, while the FACE framework does not explicitly present such a requirement. The latter one is emphasis framing, that is, the ability to explain the creative decisions taken by the system while producing the artefact. Needless to say, creativity can be understood in different ways and for as long it is not defined, it is difficult to decide the degree to which the system is creative.

The master-apprentice approach is a novel methodological contribution of the thesis. It is based on the ideology of combining a traditional machine learning approach (master) with a neural network (apprentice). The genetic algorithm, representing the traditional machine learning approach, is more interpretable and transparent than the individual fitness functions and the effect of the generation process can be traced back while the neural network operates as a black box. The apprentice can approximate creative autonomy by continuously learning from human-authored data alongside the data produced by the master. This allows it to adjust its own standards to produce artefacts beyond the master's capabilities.

This master-apprentice approach has proven its ability in making deep learning methods usable in resource-poor scenarios, as synthetic training data is produced automatically by the master to make the training of such neural models viable. While similar techniques have been studied in the field of machine learning to enhance the prediction accuracy of models, they are hardly discussed or investigated in the field of natural language generation, let alone computational creativity. Existing techniques are usually limited to simplistic data augmentation techniques (e.g. back-translation (Sennrich, Haddow, and Birch 2016)). Our approach is, however, different as the master is designed to model the problem in a higher level of granularity than a mere increase in the size of training data as in our case the training is produced in a more informed way with respect to the problem that is being modelled.

In the future, it would be interesting to investigate different types and settings of co-operations between the master-apprentice dual. This might shed light onto whether the overall system could benefit from a bi-directional communication between the master and apprentice where the master can learn from the apprentice as well. We are currently studying the viability of the master-apprentice framework in a multi-agent setting to research if having domain-expert masters (in contrast to a single generalist master) improves the quality of generated slogans. The master-apprentice approach has a great potential for being utilised in tasks outside computational cre-

ativity, for example in the context of endangered languages that have a notoriously limited amount of resources (c.f. Alnajjar et al. (2019)).

A great future direction is to tailor the NLG methods presented in this thesis to work with real-world applications such as news headline generation (Alnajjar, Leppänen, and Toivonen 2019) and online systems for poetry generation (Hämäläinen 2018b). In such applications, the system should not have an unrestricted freedom as crucial constraints should be met for the system to be considered usable and creative. For example, a news headline generator system should produce headlines that are descriptive and factual while introducing colourful language to increase their fluency and catchiness.

References

Abe, K.; Sakamoto, K.; and Nakagawa, M. 2006. A computational model of the metaphor generation process. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 937–942. Tokyo, Japan: Lawrence Erlbaum Associates.

Alnajjar, K.; Hämäläinen, M.; Chen, H.; and Toivonen, H. 2017. Expanding and weighting stereotypical properties of human characters for linguistic creativity. In *Proceedings of the 8th International Conference on Computational Creativity*, 25–32. Georgia: Georgia Institute of Technology.

Alnajjar, K.; Hämäläinen, M.; Partanen, N.; and Rueter, J. 2019. The open dictionary infrastructure for Uralic languages. In *Электронная Письменность Народов Российской Федерации*, 49–51. Russian Federation: Башкирская энциклопедия.

Alnajjar, K.; Leppänen, L.; and Toivonen, H. 2019. No time like the present: Methods for generating colourful and factual multilingual news headlines. In *Proceedings of the 10th International Conference on Computational Creativity*, 258–265. Charlotte, North Carolina, United States: Association for Computational Creativity.

Alnajjar, K. 2019. *Computational Analysis and Generation of Slogans.* Master's Thesis. University of Helsinki, Faculty of Science.

Asmis, E. 1992. Plato on poetic creativity. In Kraut, R., ed., *The Cambridge Companion to Plato.* Cambridge: Cambridge University Press. 338–364.

Attardo, S., and Raskin, V. 1991. Script theory revis(it)ed: Joke similarity and joke representation model. *Humor-International Journal of Humor Research* 4(3-4):293–348.

Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The Berkeley FrameNet project. In *Proceedings of the 17th international conference on Computational linguistics*, 86–90. Montreal, Quebec, Canada: Association for Computational Linguistics.

Bar, K.; Dershowitz, N.; and Dankin, L. 2018. Metaphor interpretation using word embeddings. *International Journal of Computational Linguistics and Applications*.

Black, M. 1962. *Models and Metaphors: Studies in Language and Philosophy*. Ithaca, New York: Cornell University Press.

Boden, M. 2004. *The Creative Mind: Myths and Mechanisms*. 11 New Fetter Lane, London: Routledge.

Boden, M. A. 2007. Creativity in a nutshell. *Think* 5(15):83–96.

Brants, T., and Franz, A. 2006. Web 1T 5-gram version 1 LDC2006T13. Philadelphia, PA, USA: Linguistic Data Consortium.

Brownell, H. H.; Michel, D.; Powelson, J.; and Gardner, H. 1983. Surprise but not coherence: Sensitivity to verbal humor in right-hemisphere patients. *Brain and language* 18(1):20–27.

Cai, Y.; Li, Y.; and Wan, X. 2018. Sense-aware neural models for pun location in texts. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 546–551. Melbourne, Australia: Association for Computational Linguistics.

Carnovalini, F., and Rodà, A. 2020. Computational creativity and music generation systems: an introduction to the state of the art. *Frontiers in Artificial Intelligence* 3:14.

Cattle, A., and Ma, X. 2018. Recognizing humour using word associations and humour anchor extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, 1849–1858. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Chalmers, D. J. 1995. Absent qualia, fading qualia, dancing qualia. In *Conscious experience*. Paderborn: Ferdinand-Schoningh. 309–328.

Christiane, F., and Brown, K. 2005. Wordnet and wordnets. In *Encyclopedia of Language and Linguistics*. Oxford: Elsevier. 665–670.

Cipresso, P.; Serino, S.; and Villani, D. 2019. *Pervasive Computing Paradigms for Mental Health*. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Buenos Aires, Argentina: Springer International Publishing.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *Proceedings of the 20th European Conference on Artificial Intelligence*, ECAI'12, 21–26. Amsterdam, The Netherlands: IOS Press.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: the FACE and IDEA descriptive models. In *Proceedings of the 2nd International Conference on Computational Creativity*, 90–95. Mexico City, Mexico: Universidad Autónoma Metropolitana, Unidad Cuajimalpa, División de Ciencias de la Comunicación y Diseño.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-FACE poetry generation. In *Proceedings of the 3rd International Conference on Computational Creativity*, 95–102. Dublin, Ireland: Association for Computational Creativity.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, Technical Report SS-08-03, 14–20. Stanford, California, USA: AAAI Press.

Colton, S. 2009. Seven catchy phrases for computational creativity research. In *Computational Creativity: An Interdisciplinary Approach*, Schloss Dagstuhl Seminar Series. Wadern, Germany: Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Colton, S. 2012. The painting fool: Stories from building an automated painter. In *Computers and Creativity*. Berlin, Heidelberg: Springer Berlin Heidelberg. 3–38.

Costa, D.; Gonçalo Oliveira, H.; and Pinto, A. M. 2015. In reality there are as many religions as there are papers – first steps towards the generation of internet memes. In *Proceedings of the 6th International Conference on Computational Creativity*, 300–307. Park City, Utah, USA: Association for Computational Creativity.

Csikszentmihalyi, M. 1997. Creativity: Flow and the psychology of discovery and invention. *HarperPerennial* 39.

D'Andrade, R., and Egan, M. 1974. The colors of emotion 1. *American ethnologist* 1(1):49–63.

Davidson, D. 1978. What metaphors mean. *Critical Inquiry* 5(1):31–47.

Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2):182–197.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Feng, Y., and Wan, X. 2019. Learning bilingual sentiment-specific word embeddings without cross-lingual supervision. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 420–429. Minneapolis, Minnesota: Association for Computational Linguistics.

Foucault, M. 1969. *L'Archéologie du Savoir*. France: Editions Gallimard.

Gagliano, A.; Paul, E.; Booten, K.; and Hearst, M. A. 2016. Intersecting word vectors to take figurative language to new heights. In *Proceedings of the 5th Workshop on Computational Linguistics for Literature*, 20–31. San Diego, California, USA: Association for Computational Linguistics.

Galván, P.; Francisco, V.; Hervás, R.; Méndez, G.; and Gervás, P. 2016. Exploring the role of word associations in the construction of rhetorical figures. In *Proceedings of the 7th International Conference on Computational Creativity*. Paris, France: Sony CSL.

Gargett, A.; Mille, S.; and Barnden, J. 2015. Deep generation of metaphors. In *The 2015 Conference on Technologies and Applications of Artificial Intelligence*, 336–343. Tainan, Taiwan: IEEE.

Gaut, B. 2012. Creativity and rationality. *The Journal of Aesthetics and Art Criticism* 70(3):259–270.

Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science* 7(2):155–170.

Gervás, P. 2001. An expert system for the composition of formal Spanish poetry. In *Applications and Innovations in Intelligent Systems VIII*. London: Springer London. 19–32.

Gervás, P. 2017. Template-free construction of rhyming poems with thematic cohesion. In *Proceedings of the Workshop on Computational Creativity in Natural Language Generation*, 21–28. Santiago de Compostela, Spain: Association for Computational Linguistics.

Giora, R. 2003. *On our mind: Salience, context, and figurative language*. Oxford University Press.

Glucksberg, S.; McGlone, M. S.; and Manfredi, D. 1997. Property attribution in metaphor comprehension. *Journal of Memory and Language* 36(1):50–67.

Gonçalo Oliveira, H.; Hervás, R.; Díaz, A.; and Gervás, P. 2017. Multilingual extension and evaluation of a poetry generator. *Natural Language Engineering* 23(6):929–967.

Gonçalo Oliveira, H. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, 11–20. Santiago de Compostela, Spain: Association for Computational Linguistics.

Hämäläinen, M., and Alnajjar, K. 2019a. Creative contextual dialog adaptation in an open world RPG. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, FDG '19, 73. New York, NY, USA: Association for Computing Machinery.

Hämäläinen, M., and Alnajjar, K. 2019b. Generating modern poetry automatically in Finnish. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5999–6004. Hong Kong, China: Association for Computational Linguistics.

Hämäläinen, M., and Alnajjar, K. 2019c. Modelling the socialization of creative agents in a master-apprentice setting: The case of movie title puns. In *Proceedings of the 10th International Conference on Computational Creativity*, 266–273. Charlotte, North Carolina, USA: Association for Computational Creativity.

Hämäläinen, M., and Honkela, T. 2019. Co-operation as an asymmetric form of human-computer creativity. case: Peace machine. In *Proceedings of the First Workshop on NLP for Conversational AI*, 42–50. Florence, Italy: Association for Computational Linguistics.

Hämäläinen, M., and Rueter, J. 2018. Development of an open source natural language generation tool for Finnish. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, 51–58. Helsinki, Finland: Association for Computational Linguistics.

Hämäläinen, M.; Partanen, N.; Alnajjar, K.; Jack, R.; and Thierry, P. 2020. Automatic dialect adaptation in finnish and its effect on perceived creativity. In *Proceedings of the 11th International Conference on Computational Creativity*. Coimbra, Portugal: Association for Computational Creativity.

Hämäläinen, M. 2018a. Harnessing NLG to create Finnish poetry automatically. In Pachet, F.; Jordanous, A.; and León, C., eds., *Proceedings of the Ninth International Conference on Computational Creativity*, 9–15. Spain: Association for Computational Creativity.

Hämäläinen, M. 2018b. Poem machine - a co-creative NLG web application for poem writing. In *Proceedings of the 11th International Conference on Natural Language Generation*, 195–196. Tilburg University, The Netherlands: Association for Computational Linguistics.

Hämäläinen, M. 2019. UralicNLP: An NLP library for Uralic languages. *Journal of Open Source Software* 4(37):1345.

Happé, F. G. 1993. Communicative competence and theory of mind in autism: A test of relevance theory. *Cognition* 48(2):101–119.

Harmon, S. 2015. FIGURE8: A novel system for generating and evaluating figurative language. In *Proceedings of the 6th International Conference on Computational Creativity*, 71–77. Park City, Utah, United States: Brigham Young University.

Havasi, C.; Speer, R.; and Holmgren, J. 2010. Automated color selection using semantic knowledge. In *2010 AAAI Fall Symposium Series*, Commonsense Knowledge, 40–45. Stanford, California, United States: AAAI Press.

He, H.; Peng, N.; and Liang, P. 2019. Pun generation with surprise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1734–1744. Minneapolis, Minnesota: Association for Computational Linguistics.

Heer, J., and Stone, M. 2012. Color naming models for color selection, image editing and palette design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, 1007–1016. New York, NY, USA: Association for Computing Machinery.

Hong, B. A., and Ong, E. 2009. Automatically extracting word relationships as templates for pun generation. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, 24–31. Boulder, Colorado: Association for Computational Linguistics.

Jordanous, A. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4(3):246–279.

Katz, A. N. 1989. On choosing the vehicles of metaphors: Referential concreteness, semantic distances, and individual differences. *Journal of Memory and Language* 28(4):486–499.

Kawakami, K.; Dyer, C.; Routledge, B.; and Smith, N. A. 2016. Character sequence models for colorful words. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1949–1954. Austin, Texas: Association for Computational Linguistics.

Kay, P., and Kempton, W. 1984. What is the sapir-whorf hypothesis? *American anthropologist* 86(1):65–79.

Kintsch, W. 1988. The role of knowledge in discourse comprehension: A construction-integration model. *Psychological review* 95(2):163.

Kintsch, W. 2000. Metaphor comprehension: A computational theory. *Psychonomic Bulletin & Review* 7(2):257–266.

Kintsch, W. 2008. *How the mind computes the meaning of metaphor.* Cambridge Handbooks in Psychology. Cambridge: Cambridge University Press. 129–142.

Kirby, J. T. 1997. Aristotle on metaphor. *American Journal of Philology* 118(4):517–554.

Klein, G.; Kim, Y.; Deng, Y.; Senellart, J.; and Rush, A. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, 67–72. Vancouver, Canada: Association for Computational Linguistics.

Koestler, A. 1964. *The act of creation.* London, United Kingdom: London Hutchinson.

Krikmann, A. 2006. Contemporary linguistic theories of humour. *Folklore: Electronic journal of folklore* 33:27–58.

Lakoff, G., and Johnson, M. 2008. *Metaphors We Live By.* University of Chicago Press.

Lamb, C., and Brown, D. G. 2019. TwitSong 3.0: towards semantic revisions in computational poetry. In *Proceedings of the 10th International Conference on Computational Creativity*, 212–219. Association for Computational Creativity.

Lerner, J. S., and Tetlock, P. E. 2003. *Bridging Individual, Interpersonal, and Institutional Approaches to Judgment and Decision Making: The Impact of Accountability on Cognitive Bias.* Cambridge Series on Judgment and Decision Making. Cambridge: Cambridge University Press. 431–457.

Li, J.; Song, Y.; Zhang, H.; Chen, D.; Shi, S.; Zhao, D.; and Yan, R. 2018. Generating classical Chinese poems via conditional variational autoencoder and adversarial training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3890–3900. Brussels, Belgium: Association for Computational Linguistics.

Lindsey, D. T., and Brown, A. M. 2006. Universality of color names. *Proceedings of the National Academy of Sciences* 103(44):16608–16613.

Littlemore, J.; Sobrino, P. P.; Houghton, D.; Shi, J.; and Winter, B. 2018. What makes a good metaphor? a cross-cultural study of computer-generated metaphor appreciation. *Metaphor and Symbol* 33(2):101–122.

Martin, J. H. 1990. *A Computational Model of Metaphor Interpretation*. Cambridge, Massachusetts, USA: Academic Press Professional, Inc.

Merrotsy, P. 2013. A note on Big-C creativity and Little-c creativity. *Creativity Research Journal* 25(4):474–476.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miller, D. W., and Toman, M. 2016. An analysis of rhetorical figures and other linguistic devices in corporation brand slogans. *Journal of Marketing Communications* 22(5):474–493.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Communications of the ACM* 38(11):39–41.

Miller, G. A. 1998. *WordNet: An electronic lexical database*. MIT press.

Misztal, J., and Indurkhya, B. 2014. Poetry generation system with an emotional personality. In *Proceedings of the 5th International Conference on Computational Creativity*, 72–81. Ljubljana, Slovenia: Association for Computational Creativity.

Mohler, M.; Tomlinson, M.; and Bracewell, D. 2013. Applying textual entailment to the interpretation of metaphor. In *Proceedings of the 2013 IEEE 7th International Conference on Semantic Computing*, ICSC '13, 118–125. USA: IEEE Computer Society.

Monroe, W.; Hawkins, R. X.; Goodman, N. D.; and Potts, C. 2017. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics* 5:325–338.

Moscovici, S. 1961. *La psychanalyse, son image et son public*. Presses universitaires de France.

Mumford, M. D. 2003. Where have we been, where are we going? taking stock in creativity research. *Creativity Research Journal* 15(2-3):107–120.

Munroe, R. 2010. Color survey results. `http://blog.xkcd.com/2010/05/03/color-survey-results/`.

Nerger, J. L.; Volbrecht, V. J.; and Ayde, C. J. 1995. Unique hue judgments as a function of test size in the fovea and at 20-deg temporal eccentricity. *JOSA A* 12(6):1225–1232.

Odbert, H. S.; Karwoski, T. F.; and Eckerson, A. 1942. Studies in synesthetic thinking: I. musical and verbal associations of color and mood. *The journal of general psychology* 26(1):153–173.

Ortony, A.; Vondruska, R. J.; Foss, M. A.; and Jones, L. E. 1985. Salience, similes, and the asymmetry of similarity. *Journal of memory and language* 24(5):569–594.

Ortony, A. 1993. *The role of similarity in similes and metaphors.* Cambridge, United Kingdom: Cambridge University Press, 2 edition. 342–356.

Özbal, G.; Pighin, D.; and Strapparava, C. 2013. BRAINSUP: Brainstorming support for creative sentence generation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 1446–1455. Sofia, Bulgaria: Association for Computational Linguistics.

Peirsman, Y.; Heylen, K.; and Geeraerts, D. 2008. Size matters: Tight and loose context definitions in english word space models. In *Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics*, 34–41. Berlin: Springer.

Pirinen, T. A.; Listenmaa, I.; Johnson, R.; Tyers, F. M.; and Kuokkala, J. 2017. Open morphology of finnish. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.

Rai, S., and Chakraverty, S. 2020. A survey on computational metaphor processing. *ACM Computing Surveys* 53(2).

Rai, S.; Chakraverty, S.; Tayal, D. K.; Sharma, D.; and Garg, A. 2019. Understanding metaphors using emotions. *New Generation Computing* 37(1):5–27.

Rapp, R. 2002. The computation of word associations: Comparing syntagmatic and paradigmatic approaches. In *COLING 2002: The 19th International Conference on Computational Linguistics.* Taipei, Taiwan: Association for Computational Linguistics.

Raskin, V. 1985. *Semantic Mechanisms of Humor.* Springer Science & Business Media.

Reece, B. B.; Van den Bergh, B. G.; and Li, H. 1994. What makes a slogan memorable and who remembers it. *Journal of Current Issues & Research in Advertising* 16(2):41–57.

Reinsch Jr., N. L. 1971. An investigation of the effects of the metaphor and simile in persuasive discourse. *Speech Monographs* 38(2):142–145.

Reiter, E. 1994. Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? In *Proceedings of the 7th International Workshop on Natural Language Generation*, 163–170. Philadelphia, PA, USA: Association for Computational Linguistics.

Reiter, E. 2018. A structured review of the validity of BLEU. *Computational Linguistics* 44(3):393–401.

Repar, A.; Martinc, M.; Žnidaršič, M.; and Pollak, S. 2018. BISLON: BISociative SLOgaN generation based on stylistic literary devices. In *Proceedings of the Ninth International Conference on Computational Creativity*, 248–255. Salamanca, Spain: Association for Computational Creativity.

Rhodes, M. 1961. An analysis of creativity. *The Phi Delta Kappan* 42(7):305–310.

Richards, I. A. 1936. *The Philosophy of Rhetoric*. London, United Kingdom: Oxford University Press.

Ritchie, G. 2005. Computational mechanisms for pun generation. In *Proceedings of the 10th European Workshop on Natural Language Generation*. Aberdeen, Scotland: Association for Computational Linguistics.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds Mach.* 17(1):67–99.

Rosen, Z. 2018. Computationally constructed concepts: A machine learning approach to metaphor interpretation using usage-based construction grammatical cues. In *Proceedings of the Workshop on Figurative Language Processing*, 102–109. New Orleans, Louisiana: Association for Computational Linguistics.

Sane, S. R.; Tripathi, S.; Sane, K. R.; and Mamidi, R. 2019. Deep learning techniques for humor detection in Hindi-English code-mixed tweets. In *Proceedings of the 10th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 57–61. Minneapolis, USA: Association for Computational Linguistics.

Searle, J. R. 1969. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 86–96. Berlin, Germany: Association for Computational Linguistics.

Setlur, V., and Stone, M. C. 2016. A linguistic approach to categorical color assignment for data visualization. *IEEE Transactions on Visualization and Computer Graphics* 22(1):698–707.

Shao, Y.; Zhang, C.; Zhou, J.; Gu, T.; and Yuan, Y. 2019. How does culture shape creativity? a mini-review. *Frontiers in psychology* 10:1219.

Sharma, G., and Bala, R. 2017. *Digital color imaging handbook*. CRC press.

Shutova, E.; Sun, L.; Darío Gutiérrez, E.; Lichtenstein, P.; and Narayanan, S. 2017. Multilingual metaphor processing: Experiments with semi-supervised and unsupervised learning. *Computational Linguistics* 43(1):71–123.

Shutova, E.; Van de Cruys, T.; and Korhonen, A. 2012. Unsupervised metaphor paraphrasing using a vector space model. In *Proceedings of COLING 2012: Posters*, 1121–1130. Mumbai, India: The COLING 2012 Organizing Committee.

Shutova, E. 2010. Automatic metaphor interpretation as a paraphrasing task. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, 1029–1037. USA: Association for Computational Linguistics.

Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, 4444–4451. Stanford, California, United States: AAAI Press.

Strapparava, C.; Valitutti, A.; and Stock, O. 2007. Automatizing two creative functions for advertising. In Cardoso, A., and Wiggins, G., eds., *Proceedings of the 4th International Joint Workshop on Computational Creativity*, 99–108. London, United Kingdom: Goldsmiths, University of London.

Su, C.; Tian, J.; and Chen, Y. 2016. Latent semantic similarity based interpretation of chinese metaphors. *Engineering Applications of Artificial Intelligence* 48:188–203.

Talman, A., and Chatzikyriakidis, S. 2019. Testing the generalization power of neural network models across NLI benchmarks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 85–94. Florence, Italy: Association for Computational Linguistics.

Terai, A., and Nakagawa, M. 2008. A corpus-based computational model of metaphor understanding incorporating dynamic interaction. In *Proceedings of The Eighteenth International Conference on Artificial Neural Networks*, ICANN '08, 443–452. Berlin: Springer.

Terai, A., and Nakagawa, M. 2012. A corpus-based computational model of metaphor understanding consisting of two processes. *Cognitive Systems Research* 19–20:30–38.

Tiedemann, J., and Scherrer, Y. 2019. Measuring semantic abstraction of multilingual NMT with paraphrase recognition and generation tasks. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, 35–42. Minneapolis, USA: Association for Computational Linguistics.

Toivanen, J.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of the 3rd International Conference on Computational Creativity*. Dublin, Ireland: Association for Computational Creativity.

Tom, G., and Eves, A. 1999. The use of rhetorical devices in advertising. *Journal of Advertising Research* 39(4):39–43.

Tomašič, P.; Papa, G.; and Žnidaršič, M. 2015. Using a genetic algorithm to produce slogans. *Informatica* 39(2):125.

Torrance, E. P. 1962. *Guiding creative talent*. Pickle Partners Publishing.

Tourangeau, R., and Sternberg, R. J. 1981. Aptness in metaphor. *Cognitive Psychology* 13(1):27–55.

Valitutti, A.; Toivonen, H.; Doucet, A.; and Toivanen, J. M. 2013. "let everything turn well in your wife": Generation of adult humor using lexical constraints. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 2, 243–248.

Valitutti, A.; Doucet, A.; Toivanen, J. M.; and Toivonen, H. 2016. Computational generation and dissection of lexical replacement humor. *Natural Language Engineering* 22(5):727–749.

Van de Cruys, T. 2020. Automatic poetry generation from prosaic text. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2471–2480. Online: Association for Computational Linguistics.

Veale, T., and Hao, Y. 2007. Comprehending and generating apt metaphors: A web-driven, case-based approach to figurative language. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, 1471–1476. Stanford, California, USA: AAAI Press.

Veale, T., and Li, G. 2012. Specifying viewpoint and information need with affective metaphors: A system demonstration of the metaphor magnet web app/service. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, 7–12. USA: Association for Computational Linguistics.

Veale, T., and Li, G. 2013. Creating similarity: Lateral thinking for vertical similarity judgments. In *Proceedings of the 51st Annual Meeting of*

the Association for Computational Linguistics, 660–670. Sofia, Bulgaria: Association for Computational Linguistics.

Veale, T. 2004. Incongruity in humor: Root cause or epiphenomenon? *Humor: International Journal of Humor Research* 17(4):419–428.

Veale, T. 2011. Creative language retrieval: A robust hybrid of information retrieval and linguistic creativity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 278–287. Portland, Oregon, USA: Association for Computational Linguistics.

Veale, T. 2012. *Exploding the creativity myth: The computational foundations of linguistic creativity.* A&C Black.

Wallas, G. 1926. *The art of thought.* J. Cape, London.

Whitman, W. 1855. *Leaves of Grass.* Modern Library Series. Random House Publishing Group.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

Wilks, Y. 1978. Making preferences more active. *Artificial intelligence* 11(3):197–223.

Winters, T.; Nys, V.; and De Schreye, D. 2019. Towards a General Framework for Humor Generation from Rated Examples. In *Proceedings of the 10th International Conference on Computational Creativity*, 274–281. Santiago de Compostela, Spain: Association for Computational Creativity.

Xiao, P., and Blat, J. 2013. Generating apt metaphor ideas for pictorial advertisements. In *Proceedings of the Fourth International Conference on Computational Creativity*, 8–15.

Yang, D.; Lavie, A.; Dyer, C.; and Hovy, E. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2367–2376. Lisbon, Portugal: Association for Computational Linguistics.

Yang, C.; Sun, M.; Yi, X.; and Li, W. 2018. Stylistic Chinese poetry generation via unsupervised style disentanglement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3960–3969. Brussels, Belgium: Association for Computational Linguistics.

Yang, Z.; Cai, P.; Feng, Y.; Li, F.; Feng, W.; Chiu, E.-Y.; and Yu, H. 2019. Generating classical Chinese poems from vernacular Chinese. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*

Processing and the 9th International Joint Conference on Natural Language Processing, 6155–6164. Hong Kong, China: Association for Computational Linguistics.

Yi, X.; Sun, M.; Li, R.; and Li, W. 2018. Automatic poetry generation with mutual reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3143–3153. Brussels, Belgium: Association for Computational Linguistics.

Yu, Z.; Tan, J.; and Wan, X. 2018. A neural approach to pun generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 1650–1660. Melbourne, Australia: Association for Computational Linguistics.

Zou, Y., and Lu, W. 2019. Joint detection and location of English puns. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2117–2123. Minneapolis, Minnesota: Association for Computational Linguistics.